

BUKU PEMBAHASAN

Bebras Indonesia

COMPUTATIONAL THINKING
CHALLENGE **2019**

PENGGALANG
(Tingkat SMP/MTs)



<https://bebras.or.id/>

Diterbitkan oleh : NBO Bebras Indonesia

Pengantar

Tantangan Bebras Indonesia adalah kompetisi yang dilaksanakan secara online dan serentak dengan memberikan soal-soal yang telah dipersiapkan dalam Workshop Bebras Internasional, pada periode Bebras Week di minggu pertama/dua bulan November.

Tantangan Bebras Indonesia dibagi dalam 3 kategori, yaitu:

1. Siaga, untuk siswa SD/MI dan yang sederajat
2. Penggalang, untuk siswa SMP/MTs dan yang sederajat
3. Penegak, untuk siswa SMA/MA/SMK/MAK dan yang sederajat.

Untuk kategori Siaga (SD/MI) diberikan 10 soal yang harus diselesaikan dalam waktu 40 menit. Untuk kategori Penggalang (SMP/MTs) dan Penegak (SMA/MA/SMK/MAK) masing-masing diberikan 12 soal yang harus diselesaikan dalam waktu 45 menit.

Tantangan Bebras Indonesia 2019 dapat berjalan lancar berkat dukungan penuh dari IPB yang menyediakan dan mengelola <https://tantanganbebras.ipb.ac.id> dan GDP Labs yang menyediakan dan mengelola <https://olympia.id> sebagai sistem aplikasi untuk lomba online. Selain dari itu juga LAPI Divusi yang membantu mengelola situs <https://bebras.or.id> serta para Koordinator Bebras Biro dan tim yang tersebar di 60 perguruan tinggi di seluruh Indonesia yang langsung berhubungan dengan para siswa dalam menyelenggarakan Tantangan Bebras Indonesia 2019.

Penyiapan soal-soal dan pengelolaan Tantangan Bebras Indonesia 2019 dilaksanakan oleh Tim Olimpiade Komputer Indonesia (TOKI), yaitu: Inggriani (ITB), Adi Mulyanto (ITB), Suryana Setiawan (UI), Julio Adisantoso (IPB), Rully Soelaiman (ITS). Yudhi Purwananto (ITS), Yugo K. Isal (UI), dan Fauzan Joko Sularto (UPJ). Penyiapan soal juga dibantu oleh Mewati Ayub (UKM), Husnul, Natalia, dan Vania Natali (Unpar), serta penyiapan buku ini dibantu oleh Aloysius Gonzaga (UAJY), Irya Wisnubhadra (UAJY), dan Risna Saragih

Bahan belajar Computational Thinking Tantangan Bebras Indonesia 2019 ini dibagi dalam tiga buku sesuai kategori, yaitu buku untuk Tingkat SD/MI (Siaga), Tingkat SMP/MTs (Penggalang), dan Tingkat SMA/MA/SMK/MAK (Penegak).

Karya ini dilisensikan di bawah lisensi Creative Commons Attribution-Non-Commercial-No Derivatives 4.0 International (CC BY-NC-SA 4.0)

Hal ini berarti Anda bebas untuk menggunakan dan mendistribusikan buku ini, dengan ketentuan:

- Attribution: Apabila Anda menggunakan materi-materi pada buku ini, Anda harus memberikan kredit dengan mencantumkan sumber dari materi yang Anda gunakan.
- Non-Commercial: Anda tidak boleh menggunakan materi ini untuk keperluan komersial, seperti menjual ulang buku ini.
- ShareAlike: Apabila Anda mengubah atau membuat turunan dari materi-materi pada buku ini, Anda harus menyebarkan kontribusi Anda di bawah lisensi yang sama dengan materi asli.

Computational Thinking

Kemampuan berpikir kreatif, kritis dan komunikasi serta kolaborasi adalah kemampuan yang paling penting dalam (*21st century learning*) pembelajaran di abad kedua-puluh-satu, di antara kemampuan-kemampuan lainnya seperti membaca, matematik, sains. Siswa zaman sekarang perlu untuk mengembangkan keterampilan berpikir, menguasai pengetahuan tentang konten dari persoalan yang dihadapi (*content knowledge*), dan mempunyai kompetensi sosial dan emosional untuk mengarungi kehidupan dan lingkungan kerja yang semakin kompleks.

Di bidang “*Computing*” (diterjemahkan ke dalam bahasa Indonesia menjadi “Informatika”), kemampuan berpikir yang perlu dikuasai sejak pendidikan dasar adalah “*Computational Thinking*” (CT). CT adalah proses berpikir untuk memformulasikan persoalan dan solusinya, sehingga solusi tersebut secara efektif dilaksanakan oleh sebuah agen pemroses informasi (“komputer”, robot, atau manusia). CT adalah sebuah metoda dan proses berpikir untuk penyelesaian persoalan dengan menerapkan:

- **Dekomposisi** dan formulasi persoalan sedemikian rupa sehingga dapat diselesaikan dengan cepat dan efisien serta optimal dengan menggunakan komputer sebagai alat bantu.
- **Abstraksi**, yaitu menyarikan bagian penting dari suatu permasalahan dan mengabaikan yang tidak penting, sehingga memudahkan fokus kepada solusi.
- **Algoritma**, yaitu menuliskan otomatisasi solusi melalui berpikir algoritmik (langkah-langkah yang terurut).
- **Pengenalan pola persoalan**, generalisasi serta mentransfer proses penyelesaian persoalan ke sekumpulan persoalan sejenis.

CT adalah sebuah cara berpikir untuk memecahkan persoalan, merancang sistem, memahami perilaku manusia. CT melandasi konsep informatika. Di dunia saat ini dimana komputer ada di mana-mana untuk membantu berbagai segi kehidupan, CT harus menjadi dasar bagaimana seseorang berpikir dan memahami dunia dengan persoalan-persoalannya yang semakin kompleks. CT berarti berpikir untuk menciptakan dan menggunakan beberapa tingkatan abstraksi, mulai memahami persoalan sehingga mengusulkan pemecahan solusi yang efektif, efisien, “fair” dan aman. CT berarti memahami konsekuensi dari skala persoalan dan kompleksitasnya, tak hanya demi efisiensi, tetapi juga untuk alasan ekonomis dan sosial.

Bagaimana Belajar Computational Thinking?

Berpikir itu dapat dipelajari dan diasah dengan berlatih, serta mengkonstruksi pola pikir berdasarkan pengalaman. *Computational Thinking* juga dapat dipelajari dengan cara berlatih menyelesaikan persoalan-persoalan yang terkait komputasi, melalui persoalan sehari-hari. Lewat latihan-latihan yang menarik, siswa menerapkan teknik yang cocok (dekomposisi, abstraksi, pengenalan pola, representasi data, algoritmik) untuk mendapatkan solusi. Setelah latihan, siswa diharapkan melakukan refleksi serta mengkonstruksi pengetahuan berpikir, kemudian membentuk pola berpikir komputasional, yang semakin lama semakin tajam, cepat, efisien, dan optimal.

Apa perbedaan ICT/TIK (Teknologi Informasi dan Komunikasi) dengan Informatika?

Sejalan dengan itu, ICT (*Information and Communication Technology*, dalam bahasa Indonesia disebut Teknologi Informasi dan Komunikasi/TIK) mulai dibedakan dengan Informatika. TIK mengarah ke penggunaan teknologi dan perangkat/gadget, sedangkan Informatika mengarah ke keilmuan dan desain produk-produk informatika baik yang nyata (piranti pintar), maupun yang abstrak seperti program aplikasi, dan algoritma.

Kemampuan TIK lebih mengarah ke penggunaan teknologi dan perangkat/*gadget*, sedangkan Informatika mengarah ke keilmuan komputasinya. Penggunaan TIK yang dimaksud bukan hanya keterampilan menggunakan gadget dan aplikasinya, tetapi juga kemampuan untuk menggunakan dan

memanfaatkan konten dengan bijak, terutama di saat kecerdasan buatan sudah banyak digunakan untuk menciptakan konten digital.

Agar bangsa Indonesia mampu bersaing dengan negara lain, anak Indonesia tidak cukup menjadi pengguna teknologi saja, melainkan harus lebih kreatif dan inovatif untuk menciptakan produk-produk TIK. Untuk ini, siswa perlu mempelajari informatika.

Kaitan Computational Thinking dengan Informatika

Di negara maju, “*Computer Science*” (yang di Indonesia diterjemahkan sebagai “Informatika”) sudah mulai diajarkan sejak usia dini di tingkat pendidikan dasar, dengan materi dan kegiatan yang dirancang dengan mengacu ke kerangka kurikulum yang disusun oleh persatuan guru-guru, asosiasi profesi informatika, perusahaan terkemuka di bidang informatika dan TIK, serta organisasi-organisasi nirlaba yang peduli terhadap perlunya edukasi tentang informatika sejak usia dini [<https://k12cs.org>].

Kerangka kurikulum Informatika tersebut mendefinisikan lima bidang pengetahuan yaitu:

- Sistem Komputer (SK),
- Jaringan Komputer (JK),
- Analisis Data (AD),
- Algoritma dan Pemrograman (AP), dan
- Dampak Sosial Informatika (DSI).

Selain pengetahuan, juga didefinisikan praktik-praktik komputasi untuk mengemas pengetahuan dan mempraktikkannya, yaitu:

- pembinaan menumbuhkan budaya komputasi,
- menciptakan artefak berkolaborasi untuk mewujudkan suatu produk TIK,
- menguji dan memperbaiki/menyempurnakan artefak TIK,
- mengenali dan mendefinisikan problem-problem komputasi,
- berkomunikasi tentang komputasi, dan
- mengembangkan serta menggunakan abstraksi.

Sejalan dengan perbaikan yang berkelanjutan, dilakukan revisi terhadap organisasi elemen pembelajaran, yang dicerminkan dari perubahan lambang kerangka kurikulum Informatika Kurikulum Merdeka sejak Tahun 2024. Perubahan yang utama adalah dalam pengemasan elemen dan praktik yang lebih menekankan kepada pembelajaran informatika tanpa komputer (“*unplugged*”, *csunplugged*, <https://www.csunplugged.org/en/>), mengingat di Indonesia masih terjadi *digital gap* dan penekanan kepada pentingnya *Computational Thinking* sebagai landasan Informatika, dan sekaligus menjadi salah satu elemen penting pembelajaran abad ke-21 yang menjadi sasaran Profil Pelajar Pancasila. *Computational Thinking* seharusnya diperkenalkan secara intuitif dan menyenangkan sejak siswa masih pada jenjang SD, sedangkan Informatika menjadi mata pelajaran wajib dimulai dari SMP. Untuk ini Bebras berusaha menjembatani dengan membuat tantangan Bebras untuk siswa SD sebagai solusi, selain penguatan *Computational Thinking* yang khusus dibutuhkan dalam disiplin ilmu Informatika.

Makna lambang Elemen Informatika yang telah direvisi:

Lambang elemen Informatika ditunjukkan pada Gambar 1 Lambang Kurikulum Informatika yang terdiri dari Lambang Elemen Informatika sebelum direvisi dan Lambang Elemen Informatika setelah direvisi.

Lambang Elemen Informatika	Lambang Elemen Informatika (revisi)
 Bangunan melambangkan pondasi, pilar utama, plafon sebagai penutup sebelum berpraktik TIK. Pada semua Fase, 8 elemen dimunculkan secara eksplisit mulai Fase A s.d. F, yang dirasakan terlalu berat.	 Lingkaran adalah lambang pembelajar sepanjang hayat Pada Fase A s.d. E hanya ada 2 elemen secara eksplisit karena dilakukan pengintegrasian elemen rumah informatika; Pada Fase F elemen BK, Literasi Digital, AD, AP.

Gambar 1 Lambang Kurikulum Informatika

Penjelasan dari Lambang Elemen Informatika yang telah direvisi adalah sebagai berikut:

Lingkaran terdalam yang merupakan inti, berisi tulisan IF (singkatan dari Informatika), dan 8 (delapan) potongan *puzzle* dengan 3 (tiga) macam ukuran yang merepresentasi bobot kedalaman dari 8 (delapan) elemen lama logo rumah IF pada CP Informatika sebelum revisi, dengan rincian:

- Empat potongan *puzzle* berukuran kecil merepresentasi TIK, SK, JKI, DSI yang dikemas menjadi **Literasi digital**, adalah *core practices* warga digital di dunia nyata.
- Tiga potongan *puzzle* berukuran besar merepresentasi *core concept* informatika (BK, AD dan AP), **area pengetahuan inti** yang perlu untuk kontribusi mewujudkan dunia nyata menjadi dunia maya, konsep dasar berpikir yang diperlukan pada era digital, menuju era *Big Data* dan *Artificial Intelligence*.
- Satu potongan *puzzle* berukuran sedang mewakili PLB pada elemen lama, yang merupakan proses/koridor/jalan dari dunia nyata menuju kreator dunia maya (*engineering process*), sekaligus merekat karya nyata, melambangkan *crosscut component* dan *engineering process* yang dilakukan oleh *Computational Artefact Creator* mencipta produk dengan bergotong royong (berkolaborasi) dan berkebinekaan global.

Lingkaran tengah berwarna kuning menunjukkan tujuan dari mata pelajaran Informatika yaitu menjadi kreator dunia maya baik dalam bentuk karya digital kreatif, maupun dalam bentuk produk komputasi yang lain.

Warna kuning dalam psikologi melambangkan ceria dan semangat. Mengajak guru dan siswa untuk menikmati mata pelajaran Informatika dengan cara menyenangkan dan bersemangat. Warna kuning dalam penerapan pembelajaran yang sukses untuk abad ke-21 berdasarkan Six Thinking Hats [de Bono, Edward] melambangkan berpikir kritis.

Tantangan Bebras

(Bebras Computational Thinking Challenge)

Situs: <https://Bebras.org>

Bebras Challenge (semula adalah *Algorithmic Challenge* kemudian menjadi *Computational Thinking Challenge*), diinisiasi oleh Prof. Valentina Dagiene dari Lithuania sejak tahun 2004, adalah kompetisi yang diadakan tahunan bagi siswa berumur 5 s.d. 18 tahun. Pada tahun 2019 diikuti oleh sekitar 2,944 juta siswa yang berasal dari 54 negara. Komunitas Bebras sebagian besar adalah para pembina IOI seperti halnya Indonesia, adalah sekumpulan akademisi yang peduli ke pendidikan informatika bagi siswa sekolah dasar dan menengah.

Bebras mengikuti perkembangan CT, lewat “*challenge*” atau tantangan yang diberikan untuk *problem solving* terkait informatika untuk kehidupan sehari-hari, yang disajikan secara menarik dan lucu. Lewat Tantangan Bebras, siswa diajak “membangun” keterampilan berpikir untuk menyelesaikan persoalan, yaitu melalui pendekatan *constructionism* yang diperkenalkan oleh Seymour Papert dari MIT. Siswa diajak belajar dengan mencoba menjawab tantangan. Jadi, tantangan Bebras bukan lomba sekedar untuk menang tetapi yang lebih penting adalah untuk belajar berpikir dan menyelesaikan persoalan. Kepada peserta yang meraih peringkat tinggi, akan diberikan sertifikat.

Tujuan Tantangan Bebras:

- Memotivasi siswa Untuk mulai tertarik ke topik-topik informatika dan memecahkan persoalan dengan menggunakan informatika
- Menstimulasi minat siswa ke informatika
- Mendorong siswa untuk menggunakan “TIK” dengan lebih intensif dan kreatif dalam aktivitas belajarnya
- Menyemangati siswa untuk berpikir lebih dalam daripada sekedar ke komputer/alatnya dan TIK.

Tantangan bebras diselenggarakan sekali setahun pada saat hampir bersamaan di seluruh dunia, sepanjang Bebras Week, yang ditetapkan pada minggu pertama/kedua bulan November.

Bebras Indonesia

Situs: <https://Bebras.or.id>

Bebras dikelola oleh pembina Pusat/Nasional TOKI. Indonesia mulai bergabung ke komunitas internasional Bebras, dan untuk pertama kali mengadakan Tantangan Bebras dalam bahasa Indonesia pada tahun 2016 secara online yang terbagi dalam 3 kategori, yaitu kategori Siaga (SD/MI), kategori Penggalang (SMP/MTs), dan kategori Penegak (SMA/SMK/MA/MAK). Penyelenggaraan Tantangan Bebras dikoordinasi oleh Perguruan Tinggi yang menjadi Mitra Bebras Indonesia, dan dapat diselenggarakan di Perguruan Tinggi koordinator atau di sekolah. Peserta ada yang menggunakan komputer, tablet, bahkan handphone.

Bagaimana Berpartisipasi pada Tantangan Bebras 2019?

Pembina Bebras Indonesia bekerja sama dengan Perguruan Tinggi mitra dengan dukungan suporter. Perguruan Tinggi (diutamakan Program Studi Informatika dan Matematika) yang berminat untuk menjadi mitra Bebras akan dihubungkan dengan Perguruan Tinggi Pembina Utama TOKI, dan sekolah yang berminat untuk mengikutsertakan siswa dapat menghubungi Perguruan Tinggi Mitra Bebras terdekat. Sebagai persiapan, Pembina Bebras tingkat Nasional juga bersedia menjadi narasumber untuk pelatihan dosen/guru yang akan akan bergabung.

Silahkan kontak via email ke info@Bebras.or.id.

Contoh-contoh soal dan latihan online dalam bahasa Indonesia dapat diakses di <https://olympia.id> dan di <https://latihan.bebas.or.id>

Untuk Latihan di arena Bebras Indonesia

Siapa pun dapat berlatih secara mandiri di situs <https://latihan.bebas.or.id> dengan langkah:

1. Buka situs di url <https://latihan.bebas.or.id> selanjutnya klik tombol "Masuk" di pojok kanan atas halaman layar
2. Sign in with Google Account atau Microsoft Account
3. Kemudian buka email Anda untuk konfirmasi.
4. Klik link yang diberikan oleh Latihan Bebras ke email Anda
5. Klik "Pendaftaran" pada Course yang Anda pilih lalu klik "Daftarkan saya" pada Course yang Anda pilih

Atau dapat juga berlatih secara mandiri di situs <https://olympia.id> dengan langkah:

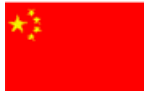
1. Akses "Create New Account", atau dari: <https://olympia.id/login/signup.php>
2. Setelah mengisi data diri secara lengkap dan password benar, selanjutnya klik "Create New Account"
3. Buka email Anda untuk konfirmasi.
4. Klik link yang diberikan oleh Olympia ke email Anda
5. Lakukan "Enroll" ke Bebras Challenge.

Negara-Negara Kontributor

Setiap soal di buku ini diberi bendera yang menandakan negara asal penyusun soal. Namun banyak pihak yang terlibat dalam mengedit, menerjemahkan, dan menyediakan material tambahan.

Bebras Indonesia berterima kasih kepada komunitas Bebras internasional karena memungkinkan kami untuk menggunakan soal-soal yang telah mereka kembangkan.

Bendera Negara Kontributor Soal-Soal pada Buku Kategori Penggalang Tantangan Bebras 2019.

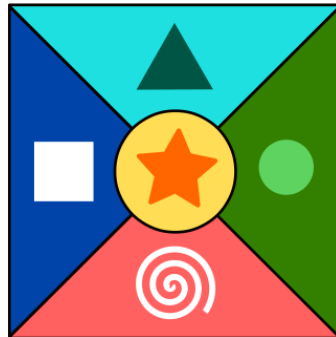


DAFTAR SOAL

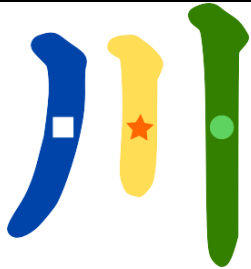
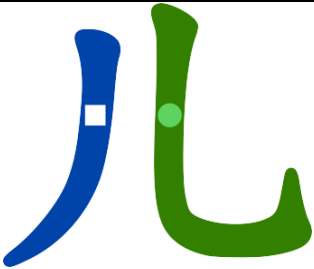
Huruf Cina Warna Warni.....	1
Parkir Mobil	4
Robot Pembersih Sampah.....	6
Pulau Jebakan	8
Mesin Bordir	11
Menata Manik-Manik	13
Mengenali Hewan	15
Alergi Kayu	17
Jembatan dan Pulau.....	20
Dalam Bahaya	22
Menara Spesial	24
Penjadwalan Penerbangan	26
Jalan Pulang Berang-Berang.....	30
Semut di Sarang	33
Pesan Nenek Moyang	35

	Huruf Cina Warna Warni	PENGGALANG (SMP) I-2019-CN-03
---	-------------------------------	-------------------------------------

Upi si berang-berang sangat tertarik untuk mempelajari struktur huruf China. Ia membuat sebuah papan warna-warni dengan simbol sebagai berikut sebagai pedoman:



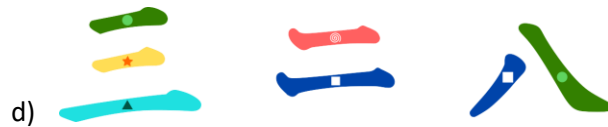
Mengacu ke papan tersebut, tulisan China dapat diekspresikan sebagai:

		
"川" Struktur kiri-tengah-kanan	"儿" Struktur kiri-kanan	"吕" Struktur atas-bawah

Tantangan: Yang mana akan menyatakan "三", "二", "八" dengan cara yang benar?

Pilihan Jawaban:

- a)   
- b)   



Jawaban:



Penjelasan:

Karakter "三" memiliki struktur "atas-tengah-bawah", dengan bagian atas berwarna biru , bagian tengah berwarna kuning , dan bagian bawah berwarna merah muda .

Karakter "二" memiliki struktur "atas-bawah", dengan bagian atas berwarna biru , dan bagian bawah berwarna merah muda .

Karakter "八" memiliki struktur "kiri-kanan", dengan bagian kiri berwarna biru , dan bagian kanan berwarna hijau .

Maka jawaban benar adalah B.

Pilihan A: Karakter "二" representasinya benar, tetapi "三" dan "八" representasinya salah.

Pilihan C: semua representasi salah.

Pilihan D: Karakter "八" representasinya benar, tetapi "二" dan "三" salah representasinya.

Ini Informatika!

Data, Struktur Data dan Representasi:

Data dapat direpresentasi menjadi berbagai bentuk, misalnya gambar, teks atau angka. Pada kasus ini, kita harus memilih deretan gambar yang akan membantu kita memecahkan persoalannya. Dengan mengidentifikasi, kita dapat memprediksi, membuat aturan dan menyelesaikan persoalan yang lebih umum.

Huruf Chinese mempunyai struktur yang seperti "*building blocks*", yang disusun dalam sebuah struktur dua dimensi. Blok ini, pada umumnya, muncul dalam bentuk "kiri-kanan", "atas-bawah" untuk membentuk sebuah huruf dua dimensi.

<http://www.littlechinesechannel.com/>

Keywords

representation, structure-pattern

Wording

- Chinese character: a way to write down a letter or word in the Chinese language.

Comments

- 2019-05-15 Syslo Maciej, syslo@ii.uni.wroc.pl, Poland, Stratis Nikolaos, nstratis@hotmail.com, Cyprus, – Augmenting colors with patterns to be suitable for color blinded students.
- 2019-05-15, Victor Schmidt (the Netherlands), victor@beverwedstrijd.nl: Did some final editing.

Graphics and Other Files

- 22019-CN-03a-Colorful Chinese Character.odt (this file)
- 2019-CN-03a_taskbody.svg, 2019-CN-03a_example01.svg, 2019-CN-03a_example02.svg, 2019-CN-03a_example03.svg, 2019-CN-03a_answerA.svg, 2019-CN-03a_answerB.svg, 2019-CN-03a_answerC.svg, 2019-CN-03a_answerD.svg, 2019-CN-03a_explanation_bottom.svg, 2019-CN-03a_explanation_center.svg, 2019-CN-03a_explanation_left.svg, 2019-CN-03a_explanation_right.svg, 2019-CN-03a_explanation_top.svg Original graphics from Yang Xing, xywzy468129@163.com replaced by graphics created by Datzko Susanne, susanne@datzko.ch, Switzerland.

Authors and Contributors (incl. Graphics)

- Yang Xing, xywzy468129@163.com, China
- Lanping Deng, 1192256382@qq.com, China
- Syslo Maciej, syslo@ii.uni.wroc.pl, Poland,
- Stratis Nikolaos, nstratis@hotmail.com, Cyprus
- Datzko Susanne, susanne@datzko.ch, Switzerland, Changes to the graphics

License

Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



Parkir Mobil

PENGGALANG
(SMP)
I-2019-DE-02

Pada sebuah tempat parkir, mobil dapat diparkir pada ruang yang disediakan, atau di depannya seperti ditunjukkan pada gambar berikut:



Mobil yang diparkir di depan ruang parkir dapat didorong maju atau mundur agar mobil yang terhalang dapat keluar. Misalnya, pada gambar di atas, mobil A tidak terhalang sehingga dapat keluar. Mobil C terhalang mobil P. Mobil M harus didorong sebelum L atau K dapat keluar.

Tantangan: Mobil apa yang membutuhkan dua mobil lainnya didorong sebelum bisa keluar?

Pilihan Jawaban:

- a) Mobil B
- b) Mobil C
- c) Mobil F
- d) Mobil G
- e) Mobil H
- f) Mobil I
- g) Mobil K
- h) Mobil L

Jawaban yang benar adalah: f) Mobil I.

Mobil I terhalang oleh mobil N. Kita tidak dapat mendorong mobil N supaya mobil I dapat meninggalkan tempat parkirnya. Maka, mobil O harus didorong ke kiri atau mobil M harus didorong ke kanan agar mobil N dapat keluar dari tempat parkirnya.

Tak ada mobil lain yang untuk meninggalkan tempatnya harus mendorong 2 mobil: mobil A, D, E, J dan Q dapat langsung meninggalkan tempat parkir. Mobil B, C, F, G, H, K dan L dapat meninggalkan tempat parkir jika mobil P, O, N, atau M didorong.

Ini Informatika!

Soal ini berkaitan dengan dua aspek penting informatika:

1. Pencarian menyeluruh, satu per satu mobil, dengan algoritma “brute-force” untuk mengecek setiap mobil apakah memenuhi kriteria, i.e. “mobil mana yang dapat keluar setelah 2 mobil lainnya didorong” https://en.wikipedia.org/wiki/Brute-force_search
2. Algoritma parkir mobil tanpa awak, yang makin banyak dibuat saat ini: https://en.wikipedia.org/wiki/Automatic_parking

Banyak penelitian dilakukan dalam bidang transportasi publik dan sistem perparkiran mobil, yang melayani mobil tanpa awak. Salah satu keuntungannya adalah membuat sistem menjadi sangat efisien.

Authorship

- 2019-03-14 Kirsten Schlüter (DE), kirsten.schlueter@sonnwendjoch.de: Task Proposal
- 2019-05-14 Sophie Koh (Singapore, sophie.k@simcc.org), Inggriani Liem (Indonesia, inge@informatika.org), Marios Choudary (Pakistan, marios.choudary@gmail.com): Simplified introductory text. Updated It's informatics text to link the task to the concepts of exhaustive search and autonomous parking. Updated wording part (delete last sentence about "to push away a car"). Updated title.
- 2019-05-15 Christian Datzko, christian.datzko@informatik-biber.ch, Switzerland: Review
- 2019-05-16 Identified ambiguity in original graphics and final language edits. Thomas Naughton (Ireland) tomn@cs.nuim.ie

License

Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking. This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <http://creativecommons.org/licenses/by-sa/4.0/>



Robot Pembersih Sampah

PENGGALANG
(SMP)
I-2019-DE-04

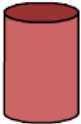
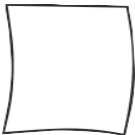
Setelah acara pentas musik, banyak sampah di lapangan. Robot Pembersih mengambil sampah yang ditinggalkan penonton:



Robot pembersih akan bergerak mengambil sampah paling dekat. Kemudian ia bergerak dan mengambil sampah paling dekat berikutnya. Robot mengulang langkah sampai semua sampah dipungutnya.

Tantangan: Benda apa yang dipungut paling akhir oleh robot?

Pilihan Jawaban:

A	B	C	D
			



Jawaban yang tepat: B.

Gambar menunjukkan rute yang diambil robot sesuai dengan instruksi. Dimulai dengan mengambil



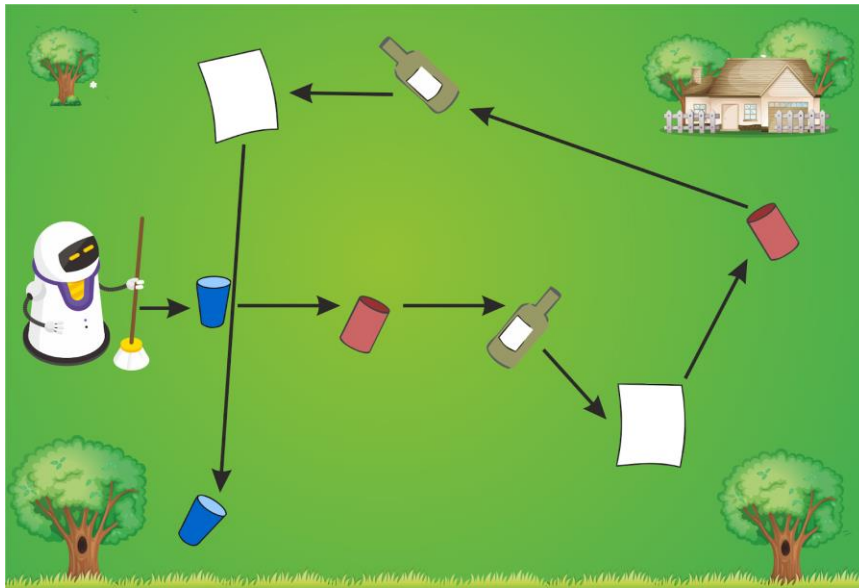
karena itu adalah sampah terdekat. Kemudian pindah ke



karena paling dekat dan

mengambilnya. Ini mengikuti rute yang ditunjukkan pada gambar di bawah ini. Sampah yang diambil

terakhir adalah cangkir kertas



Ini Informatika!

Perilaku si robot dikendalikan oleh program komputer, yang dibangun di dalam robot. Program komputer terdiri dari instruksi yang ditulis dalam bahasa yang dapat dimengerti oleh sang komputer. Robot memiliki kamera dan sensor yang memungkinkan mereka untuk mengidentifikasi objek di lingkungan mereka dan menghitung jarak ke objek-objek ini. Program kontrol robot memproses input dari sensor-sensor ini dan memungkinkan robot untuk berperilaku secara otonom, yaitu tanpa campur tangan manusia.

Robot pengumpul sampah melakukan pekerjaan yang bisa berbahaya bagi manusia. Demikian pula, robot otonom dapat membantu dalam banyak skenario berbahaya lainnya juga, seperti bencana yang disebabkan oleh halangan atau cuaca.

Authorship

- 2019-03-01 Task proposal: Michael Weigend (Germany), mw@creative-informatics.de
- 2019-05-15: Eslam AbdElAal (Egypt) (eslam@aast.edu), Georgios Fesakis (Greece) (gfesakis@aegean.gr), Vipul Shah (India), v.shah@tcs.com

License

Copyright © 2019 Bebras – International Contest on Informatics and Computer Fluency.

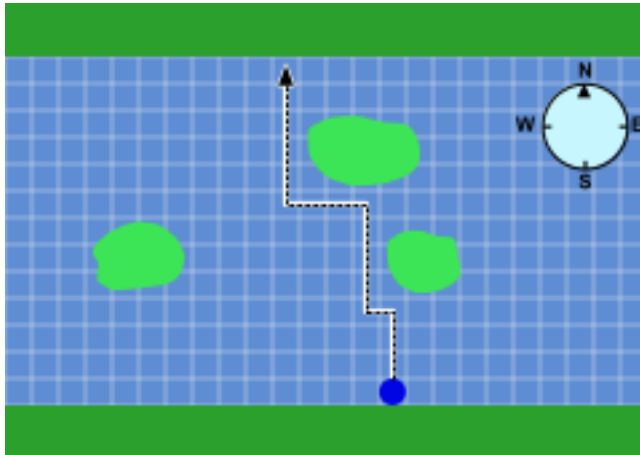
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <http://creativecommons.org/licenses/by-sa/4.0/>



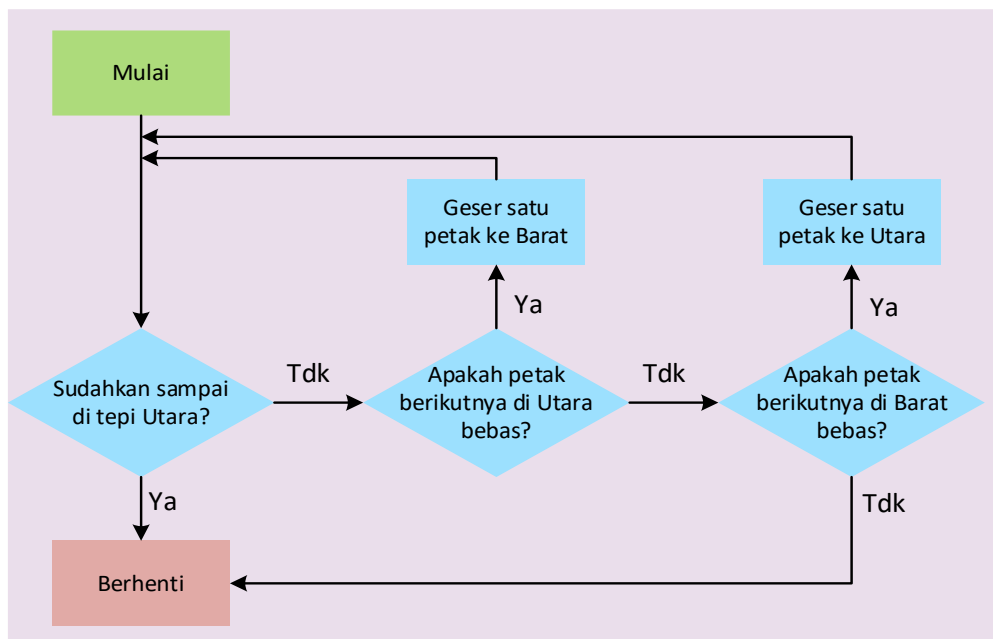
Pulau Jebakan

PENGGALANG
(SMP)
I-2019-DE-05

Sebuah kapal tanpa awak bergerak dari selatan menuju ke utara sungai. Kapal harus melewati pulau-pulau yang ada pada sungai tersebut. Kapal bergerak dari satu petak ke petak berikutnya. Kapal hanya dapat bergerak jika tidak terhalang pulau.



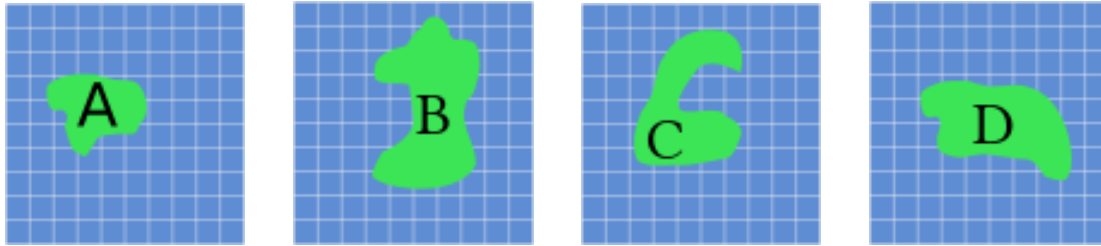
Kapal tersebut bergerak mengikuti instruksi sebagai berikut:



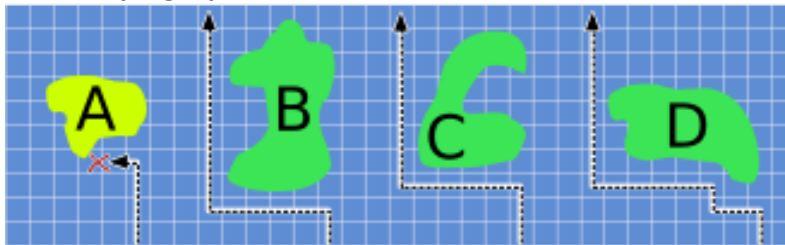
Seorang bajak laut membuat sebuah pulau buatan, yang akan dipakai menjebak kapal tanpa awak tersebut. Jika kapal mendekati pulau buatan tersebut, maka kapal akan berhenti ("stuck") sebelum mencapai tepi utara sungai.

Tantangan: Pulau yang mana dapat menjebak kapal tanpa awak tersebut?

Pilihan Jawaban:



Jawaban yang tepat adalah: A.



Jika kapal tanpa awak mendekati pulau yang di kiri seperti pada gambar, maka akan berhenti tanpa mencapai tepi utara sungai sebab kapal terhalang pulau sehingga tak dapat pergi ke utara atau ke barat. Kapal tak mungkin terhalang pulau lainnya karena saat mendekati pulau selalu bisa bergerak ke arah barat dan melanjutkan perjalanan ke utara.

Catatan: dari algoritma, kapal hanya akan bergerak ke utara dan ke barat.



Ini Informatika!

Flowchart atau diagram alir adalah diagram yang merepresentasikan algoritma, aliran kerja atau proses. Kota menunjukkan sebuah proses, dan panah menunjukkan aliran dari satu proses ke proses lainnya. Sebuah diagram alir yang baik dapat dengan jelas proses dan menunjukkan aliran, atau percabangan dari sebuah pengambilan keputusan.

Diagram alir yang mengendalikan gerakan kapal tanpa awak akan bekerja baik untuk pulau yang tepiannya membulat. Tapi diagram alir tsb tidak memperhitungkan kondisi yang menyebabkan kapal tidak dapat mencapai tepian utara karena terhalang pulau. Artinya diagram alir tersebut mempunyai kelemahan, yaitu pelanggaran keamanan (*security breach*), yang dapat dimanfaatkan oleh orang jahat. Pemrogram komputer berusaha mengembangkan perangkat lunak yang dapat berjalan dengan baik dalam kondisi apapun. Misalnya sebuah halaman web yang menerima input akan memeriksa data yang dimasukkan dengan cermat dan menolak input yang membahayakan aplikasi atau komputernya.

Authorship

- 2019-03-01 Task proposal: Michael Weigend (Germany), mw@creative-informatics.de
- 2019-05-14 Ezra Templonuevo (Philippines), eitemplonuevo@gmail.com and Aliaksei Tolstikau (Belarus), tolstikov@bsu.by

License

Copyright © 2019 Bebras – International Contest on Informatics and Computer Fluency.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <http://creativecommons.org/licenses/by-sa/4.0/>



Mesin Bordir

PENGGALANG
(SMP)
I-2019-KR-05

Upik si berang-berang mencoba untuk menyulam menggunakan pola dan sebuah mesin bordir. Mesin bordir diperintah dengan perintah: **OUT(cc) - IN(dd)** di mana **cc** dan **dd** adalah posisi jarum pada grid. Misalnya **OUT(B2)-IN(A3)** adalah perintah untuk mengeluarkan jarum dari posisi B2, dan memasukkan jarum di posisi A3. Hasil dari dua perintah **OUT(E6)-IN(G8);OUT(E2)-IN(E4)** adalah sulaman sebagai berikut.

	1	2	3	4	5	6	7	8	9	10
A	○	○	○	○	○	○	○	○	○	○
B	○	○	○	○	○	○	○	○	○	○
C	○	○	○	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○	○	○	○
E	○	○	○	○	○	○	○	○	○	○
F	○	○	○	○	○	○	○	○	○	○
G	○	○	○	○	○	○	○	○	○	○
H	○	○	○	○	○	○	○	○	○	○
I	○	○	○	○	○	○	○	○	○	○
J	○	○	○	○	○	○	○	○	○	○

Tantangan: Perintah apa yang harus dituliskan untuk menghasilkan sulaman sebagai berikut?

	1	2	3	4	5	6	7	8	9	10
A	○	○	○	○	○	○	○	○	○	○
B	○	○	○	○	○	○	○	○	○	○
C	○	○	○	○	○	○	○	○	○	○
D	○	○	○	○	○	○	○	○	○	○
E	○	○	○	○	○	○	○	○	○	○
F	○	○	○	○	○	○	○	○	○	○
G	○	○	○	○	○	○	○	○	○	○
H	○	○	○	○	○	○	○	○	○	○
I	○	○	○	○	○	○	○	○	○	○
J	○	○	○	○	○	○	○	○	○	○

Pilihan Jawaban:

- a) OUT(H2)-IN(C2);OUT(H9)-IN(C9);OUT(C9)-IN(C2);OUT(H9)-IN(C2)
- b) OUT(C2)-IN(H9);OUT(H2)-IN(C9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)
- c) OUT(H9)-IN(C9);OUT(H9)-IN(H2);OUT(C2)-IN(H2);OUT(C9)-IN(H2)
- d) OUT(C2)-IN(C9);OUT(H2)-IN(H9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

Jawaban yang tepat adalah: b) OUT(C2)-IN(H9);OUT(H2)-IN(C9);OUT(C2)-IN(H2);OUT(C9)-IN(H9)

Untuk menghasilkan pola seperti gambar, kita perlu untuk menjalankan 4 perintah, satu perintah untuk setiap garis, secara random. Perintahnya adalah:

- OUT(C2)-IN(H9) or OUT(H9)-IN(C2)
- OUT(H2)-IN(C9) or OUT(C9)-IN(H2)
- OUT(C2)-IN(H2) or OUT(H2)-IN(C2)
- OUT(C9)-IN(H9) or OUT(H9)-IN(C9)

Pilihan yang mengandung 4 perintah tersebut adalah B.

Pilihan A salah sebab mengandung perintah OUT(C9)-IN(C2) yang membuat garis yang tidak perlu. Perintah OUT(H2)-IN(C9) atau OUT(C9)-IN(H2) juga tidak ada.

Pilihan B benar.

Pilihan C salah sebab mengandung perintah OUT(H9)-IN(H2) yang membuat garis yang tidak diperlukan. Demikian juga dengan perintah OUT(C9)-IN(H9) atau OUT(H9)-IN(C9) sebab meleset.

Pilihan D salah sebab mengandung perintah OUT(C2)-IN(C9) dan OUT(H2)-IN(H9) yang membuat garis yang tidak diperlukan. Perintah OUT(C2)-IN(H9) atau OUT(H9)-IN(C2) dan OUT(H2)-IN(C9) atau OUT(C9)-IN(H2) juga meleset.

Ini Informatika!

Algoritma menjelaskan langkah-langkah untuk menyelesaikan sebuah tugas. Selain untuk dijalankan komputer, algoritma yang ditulis juga dapat menyelesaikan persoalan sehari-hari. Soal pada tantangan ini menunjukkan bagaimana algoritma dapat dipakai untuk membuat pola sebuah mesin bordir.

Authorship

- Tae-hun Kim (South Korea), gtranu@gmail.com, Task Proposal
- Hongjin Yeh (South Korea) hjyeh@ajou.ac.kr : Task Modified
- Syslo Maciej, syslo@ii.uni.wroc.pl, Poland
- Stratis Nikolaos, nstratis@hotmail.com, Cyprus
- Susanne Datzko, susanne@datzko.ch, Switzerland

License

Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking.

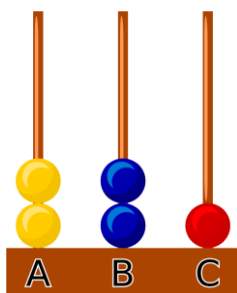
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



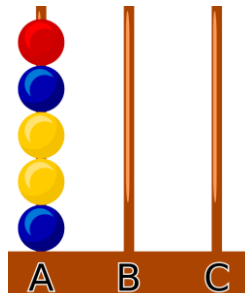
Menata Manik-Manik

PENGKALANG
(SMP)
I-2019-MY-04

Perhatikan manik-manik yang terletak dalam tongkat A, B, dan C.



Gambar I



Gambar II

Kita hanya dapat memindahkan satu manik-manik dari satu tongkat ke tongkat lainnya. Satu kali memindahkan dihitung sebagai 1 langkah.

Tantangan: Berapa pemindahan yang dibutuhkan (yang paling minimum) untuk mengatur manik-manik pada gambar I di atas untuk menjadi seperti pada Gambar II?

Pilihan Jawaban:

- a) 4
- b) 5
- c) 6
- d) 7
- e) 8

Jawaban:

Jawaban yang tepat adalah: d) 7.

Pada susunan yang diinginkan, manik-manik biru terletak paling bawah, maka 2 buah manik-manik kuning harus dipindahkan ke C (jika kita pindahkan ke B, manik-manik kuning akan menghalangi manik-manik biru. Jumlah perpindahan: 2

Kemudian, kita akan memindahkan manik-manik ke A sesuai urutan warna yang diinginkan: biru, kuning, kuning, biru, dan merah. Jumlah perpindahan: 5.

Total perpindahan: 7

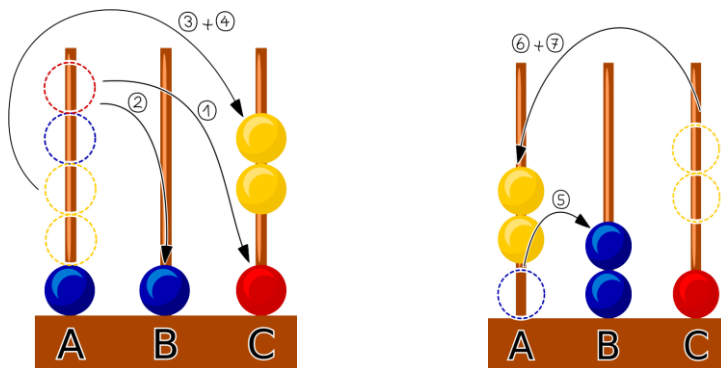
Cara lain menyelesaikan soal ini adalah dengan cara terbalik. Karena manik-manik merah harus paling atas, maka harus dipindah ke C, berarti sudah pada posisi awalnya. Jumlah perpindahan: 1

Manik-manik biru yang kemudian paling atas dapat dipindahkan ke B, yaitu posisi asalnya. Jumlah perpindahan: 1

Manik-manik kuning semula pada A, dan manik-manik biru harus dipindahkan sebelumnya sehingga manik-manik kuning harus dipindahkan ke C untuk sementara (jika dipindahkan ke B, manik-manik kuning akan menghalangi biru). Jumlah perpindahan: 2

Sekarang, manik-manik biru pada A dapat dipindah keB, dan kedua manik-manik kuning dapat dikembalikan ke A. Jumlah perpindahan: 3

Total perpindahan: 7



Ini Informatika!

Solusi dapat diperoleh dengan mudah dengan menganggap posisi target sebagai awal dan melakukan *backtracking*. Strategi “*brute force*” dengan mencoba satu per satu juga dapat menghasilkan solusi, yang biasanya dipakai dengan memanfaatkan kecepatan komputer (secara manual akan lama)

Authorship

- Nik Ahmad Faris Bin Nik Norhisham (MY), faris@ardentedu.com
- Ezra Templonuevo (Philippines), eitemplonuevo@gmail.com,
- Aliaksei Tolstikau (Belarus), tolstikov@bsu.by
- Susanne Datzko, Switzerland, susanne@datzko.ch

License

Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <http://creativecommons.org/licenses/by-sa/4.0/>



Mengenali Hewan

PENGGALANG
(SMP)
I-2019-PK-07

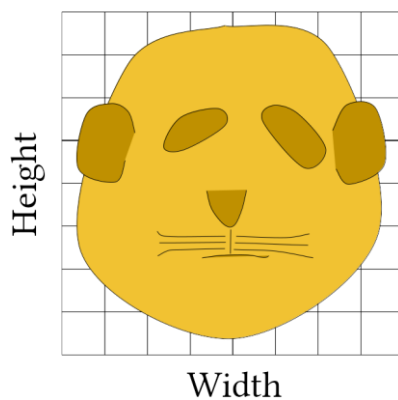
Raja Bebras ingin mendapatkan informasi tentang semua hewan penghuni hutan kekuasaannya. Untuk itu, ia mengundang semua penghuni hutan untuk datang ke istana. Sang Raja membeli sebuah mesin pengenalan hewan, yang bekerja berdasarkan wajah. Mesin tersebut bekerja mengenali wajah berdasarkan aturan sebagai berikut:

Karakteristik	Kelinci	Berang-berang	Beruang	Kucing
Tinggi telinga	$\frac{1}{2}$ tinggi kepala	$\frac{1}{4}$ tinggi kepala	$\frac{1}{4}$ tinggi kepala	$\frac{1}{2}$ tinggi kepala
Lebar kumis*	Lebar kepala	$\frac{1}{2}$ lebar kepala	$\frac{1}{2}$ lebar kepala	Lebar kepala
Lebar kepala	$\frac{1}{2}$ tinggi kepala	$\frac{1}{2}$ tinggi kepala	Tinggi kepala	Tinggi kepala

*Catatan:

- (1) lebar kumis diukur dengan menambahkan bagian kiri dan bagian kanan.
- (2) semua ukuran adalah lebar maksimum dan tinggi maksimum.

Tantangan: Hewan yang harus diidentifikasi adalah sebagai berikut:



Pilihlah salah satu dari 4 hewan yang dituliskan pada tabel di atas.

Pilihan Jawaban:

- Kelinci
- Berang-berang
- Beruang
- Kucing

Jawaban:

Jawaban yang tepat adalah c) Beruang.

Gambar tersebut memiliki karakteristik Beruang, yang dikenali oleh mesin karena:

- lebar kepala sama dengan tinggi kepala
- panjang kumis adalah setengah dari lebar kepala
- telinga adalah seperempat dari tinggi kepala

Tidak satu pun dari tiga hewan lainnya yang memiliki semua karakteristik ini secara bersamaan.

Ini Informatika!

Ini adalah aplikasi khas Machine Learning yang merupakan domain Kecerdasan Buatan untuk mengenali gambar. Machine Learning adalah bidang yang sangat penting dalam Ilmu Komputer, mengingat banyaknya aplikasi yang sedang dikembangkan karena kemajuan dalam daya komputasi. Contoh dari aplikasi yang sangat populer saat ini adalah Autonomous Driving, di mana algoritma Machine Learning dilatih untuk mengenali garis dan tanda di sepanjang jalan serta semua benda yang mungkin ditemui oleh mobil. Dalam tugas Bebras ini, kami memiliki mesin yang sudah terlatih untuk mengklasifikasikan berbagai jenis hewan dan kami ingin siswa menerapkan algoritma klasifikasi dasar yang dilakukan oleh algoritma Machine Learning tersebut.

https://en.wikipedia.org/wiki/Machine_learning

https://en.wikipedia.org/wiki/Self-driving_car

Authorship

- 2019-04-09 Marios Omar Choudary (Pakistan), marios.choudary@gmail.com
Task Proposal and graphics.

License

Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



Alergi Kayu

PENGGALANG
(SMP)
I-2019-SI-02

Tahukah anda, bahwa ternyata para berang-berang gemar makan kayu. Tapi untuk beberapa berang-berang, makan beberapa tipe kayu tertentu akan menyebabkan sakit. Ana si berang-berang ingin memasak makanan untuk pesta, dan tidak ingin dirinya dan teman-teman yang diundangnya akan sakit. Setiap masakan disiapkan dari satu macam kayu, dan berang-berang dengan senang hati akan berbagi satu jenis masakan yang sama dengan berang-berang lainnya asalkan tidak menyebabkan sakit.

Ana mempunyai daftar bebras dan kayu yang dapat dimakan tanpa menyebabkan sakit sebagai berikut:

Nama	Kayu yang tidak menyebabkan sakit
Ann	Willow, Oak, Ash, Maple
Benjamin	Willow, Oak, Poplar
Cecil	Oak
Danny	Ash, Birch
Emma	Willow, Maple, Birch
Fred	Oak, Ash
George	Poplar, Maple



Anna tak ingin menghidangkan 6 macam masakan dengan 6 macam kayu tersebut.

Tantangan: Berapa **jumlah minimum** masakan yang dapat dihidangkan oleh Anna sehingga tak ada satupun berang-berang akan sakit?

Pilihan Jawaban:

- 1
- 2
- 3
- 4
- 5
- 6

Jawaban:

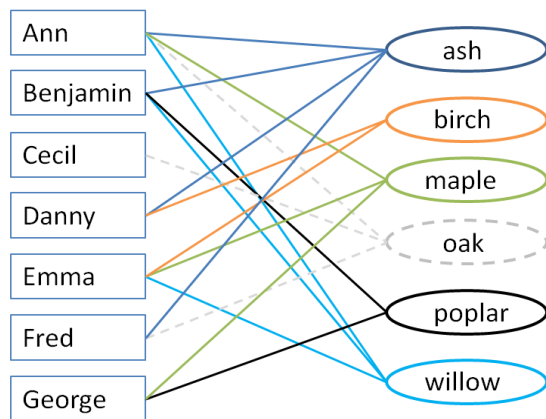
Jawaban yang tepat adalah c) 3.

Ann pasti harus membuat hidangan dari oak untuk Cecil. Hal ini juga membuat Ann, Benjamin, dan Fred senang. Tidak ada hidangan yang dapat dinikmati bersama oleh tiga berang-berang yang tersisa, sehingga Ann harus menyiapkan minimal dua hidangan lagi. Ann dapat membuat ash dan maple, poplar dan birch, atau maple dan birch.

Ini Informatika!

Soal ini adalah contoh klasik dari sebuah bipartite graph
(https://en.wikipedia.org/wiki/Bipartite_graph)

Di sini partisi pertama berisi set dari barang-barang dan partisi kedua adalah jenis kayu



Persoalan ini juga dikenal sebagai persoalan optimasi jalur kereta, dimana masukan adalah jadwal dari kereta dan perhentian, dan tujuan persoalan adalah menemukan set stasiun kereta paling optimal sehingga setiap kereta paling tidak mengunjungi satu dari stasiun kereta yang dipilih.

Persoalan ini juga dikenal sebagai Dominating Set Cover Problem. Bayangkan terdapat enam set objek yang merepresentasikan hidangan yang berbeda. Setiap set berisi barang-barang yang menyukai hidangan tersebut. Tugas yang dilakukan adalah memilih jumlah set hidangan terkecil dimana gabungan dari set-set tersebut (semua barang-barang yang menyukai hidangan tertentu) mengandung semua elemen (semua barang-barang). Dengan kata lain, tugas yang harus dilakukan adalah mencari kombinasi set dengan jumlah terkecil yang dapat mencakup semua elemen.

Seperti halnya kita mengekspresikan tugas mengenai barang-barang dan hidangan dalam bentuk set, banyak tugas atau persoalan lain yang dapat diterjemahkan dengan cara yang sama atau menyerupai set.

Persoalan Set cover dikenal menjadi salah satu dari persoalan yang paling sukar dalam ilmu komputer. Kita tidak dapat mengetahui secara pasti algoritma yang efisien untuk memecahkan persoalan tersebut. Satu-satunya solusi umum yang dapat diterapkan adalah dengan mencoba segala kemungkinan--bahkan dengan jumlah set yang kecil dapat memiliki jumlah yang sangat besar untuk semua kombinasi set yang memungkinkan, dimana dengan 240 hidangan dapat memberikan jumlah kombinasi set yang sekiranya sama dengan jumlah atom di seluruh jagat raya.

Lalu mengapa kita dapat memecahkan persoalan ini? Pertama, jumlah set relatif kecil. Kedua, hal ini dipermudah karena Cecil memiliki opsi yang terbatas. Akhirnya kita menggunakan logika sederhana yang dapat bekerja pada data ini secara spesifik, namun tidak selalu dapat membantu kita pada persoalan yang lainnya.

https://en.wikipedia.org/wiki/Set_cover_problem

Authorship

- 2019-04-09 Janez Demšar (Slovenia), janez.demsar@fri.uni-lj.si: Task Proposal
- In the Bebras Workshop 2019:
- 2019-05-15 Graeme Buckie (Australia), graeme.buckie@csiro.au, Rostyslav Shpakovych (Ukraine), rshpakovych@gmail.com, Javier Bilbao (Spain), javier.bilbao@ehu.es

License

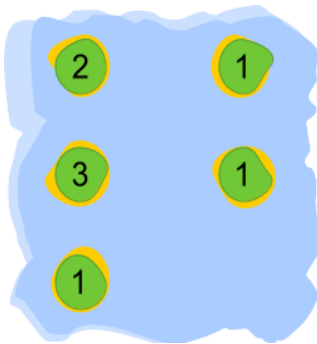
Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking.
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



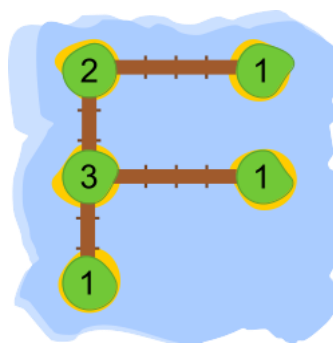
Jembatan dan Pulau

PENGGALANG
(SMP)
I-2019-VN-05

Peta berikut menunjukkan pulau-pulau yang digambar sebagai lingkaran. Untuk bepergian dari satu pulau ke pulau lainnya, bebras membutuhkan jembatan. Angka yang dituliskan pada setiap pulau menunjukkan berapa banyak jembatan yang menghubungkan ke pulau tersebut. Jembatan hanya dapat dibangun dengan arah horizontal atau vertikal. Ketika jembatan sudah dibangun, keseluruhan jembatan harus memungkinkan bebras bepergian dari satu pulau ke pulau manapun. Salah satu contoh, jika dipunyai peta seperti pada Gambar A, maka jembatan yang dibangun seperti pada Gambar B di sebelahnya.

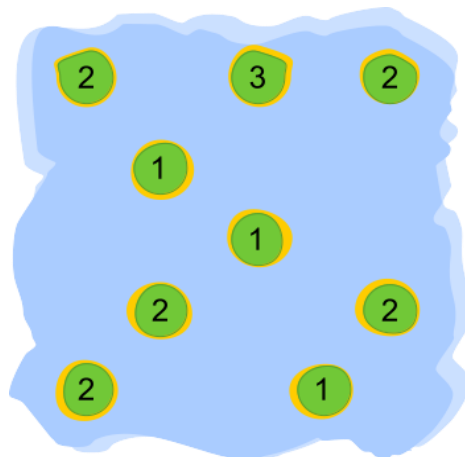


Gambar A



Gambar B

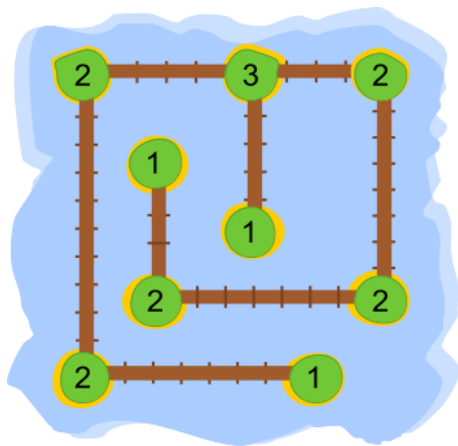
Tantangan: Berapa jembatan yang perlu dibangun untuk pulau-pulau sebagai berikut?



- a) 7
- b) 8
- c) 9
- d) 10

Jawaban:

Jawaban yang benar adalah b) 8.



Dalam setiap sistem jembatan yang lengkap, setiap jembatan akan dihitung berdasarkan dua pulau yang dihubungkan. Maka untuk melengkapi peta tersebut, jumlah jembatan seharusnya setengah dari total angka pada peta, dimana $(2+3+2+1+1+2+2+2+1)/2=8$. Untuk melengkapi jawaban maka harus ditunjukkan bahwa sistem jembatan yang lengkap dapat dibuat pada peta yang diberikan.

Ini Informatika!

Pada teori graf, salah satu cabang matematika, objek direpresentasi dengan bulatan yang dinamakan simpul, dua buah simpul dihubungkan oleh garis. Derajat dari simpul adalah banyaknya garis yang menghubungkan dengannya yang pada soal di atas menyatakan jembatan penghubung antar pulau. Jumlah derajat pada sebuah graf sama dengan dua kali jumlah garis karena setiap garis penghubung mempunyai dua buah titik ujung.

Ini dapat diperluas ke lemma jabat tangan yang menyatakan bahwa setiap graf tak berarah berhingga memiliki jumlah simpul genap dengan derajat ganjil (jumlah tepi yang menyentuh simpul). Dalam istilah yang lebih sehari-hari, dalam sekelompok orang yang beberapa di antaranya berjabat tangan, sejumlah orang pasti telah berjabat tangan dengan jumlah ganjil.

Authorship

- Vu Van Luan and Le Anh Vinh, Vietnam, yuvanluanftu.k50@gmail.com, leanhvinh@gmail.com
- Ezra Templonuevo (Philippines), ejtemplonuevo@gmail.com,
- Aliaksei Tolstikau (Belarus), tolstikov@bsu.by,
- Susanne Datzko (Switzerland), susanne@datzko.ch

License

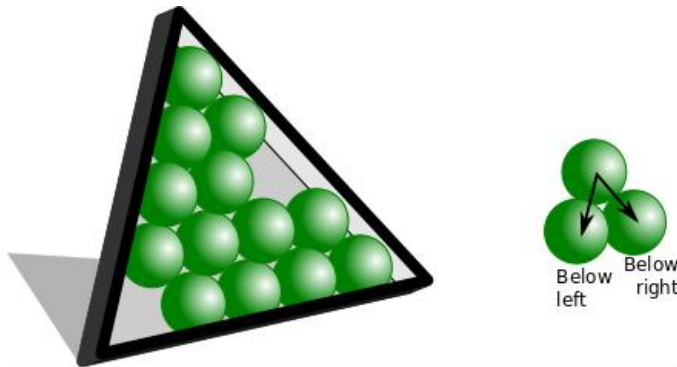
Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking. This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



Dalam Bahaya

PENGGALANG
(SMP)
I-2019-BE-02

Bebras harus menaruh 13 buah bola dalam sebuah wadah berbentuk segitiga, seperti ditunjukkan pada gambar. Jika kita mengangkat ujung atas wadah seperti gambar, bola akan bergulir ke bawah karena adanya celah (rongga).



Dikatakan bahwa bola dalam keadaan bahaya jika salah satu kondisi berikut benar:

- Ada setidaknya satu celah ke kiri atau kanan tepat di bawah bola itu.
- Setidaknya ada satu bola yang 'dalam bahaya' ke kiri atau ke kanan tepat di bawah bola itu.

Tantangan: Berapa banyak bola di kotak tersebut yang TIDAK dalam keadaan bahaya?

Pilihan Jawaban:

- a) 5
- b) 6
- c) 7
- d) 8
- e) 9

Jawaban:

Jawaban yang benar adalah d) 8.

Salah satu cara untuk menemukan jawabannya adalah sebagai berikut:

1. Mulai dari baris paling bawah dan tandai semua bola pada baris tersebut.
2. Lanjut ke baris di atasnya.
3. Tandai semua bola pada baris tersebut yang mempunyai bola ditandai pada baris di bawahnya.
4. Ulangi langkah 2 dan langkah 3 until sampai semua baris diproses.
5. Pada akhir proses ini, semua bola yang ditandai adalah yang tidak dalam bahaya.

Langkah tersebut diilustrasikan pada gambar di bawah. Mulai dari baris ke-4 dst, tidak ada bola yang bisa ditandai



Ini Informatika!

Ada dua kondisi yang mengindikasikan bola 'dalam bahaya'. Yang pertama bisa dicek langsung. Untuk yang kedua, kita perlu tahu apakah bola pada baris dibawahnya 'dalam bahaya'. Meskipun kelihatannya seperti definisi “melingkar” (sirkular), kita bisa mulai dari baris paling bawah. Bola-bola pada baris terbawah tidak memiliki baris di bawahnya, sehingga dapat disimpulkan bahwa semuanya tidak dalam bahaya. Setelah semua bola pada baris paling bawah ditandai, kita dapat memberi label pada baris di atasnya. Dengan cara ini, secara sistematis, pada setiap baris, setiap bola diberi label mulai dari bawah ke atas.

Teknik yang digunakan dalam soal ini disebut rekursi. Biasanya, rekursi menggunakan konsep yang dideskripsikan dengan menggunakan deskripsinya sendiri: untuk menjelaskan apa artinya 'dalam bahaya' kita sudah menggunakan kata 'dalam bahaya' dalam penjelasan - seolah-olah kita sudah tahu apa artinya.

Contoh lain dari definisi rekursif: jika saya meminta Anda untuk menjelaskan siapa leluhur Anda, Anda dapat menjawab 'ibu saya, ayah saya, nenek saya, nenek saya yang lain dan dua kakek saya, dan juga kakek buyut saya dan kakek buyut saya, dan seterusnya ...'. Komputer memiliki kesulitan memahami 'dan seterusnya...' sebab tidak jelas. Maka programmer akan menggunakan definisi rekursif, sebagai berikut: **Nenek moyang** Anda adalah: orang tua Anda dan **nenek moyang mereka**. Bagi Anda ini mungkin deskripsi ini aneh, tetapi bagi komputer ini menjadi sangat jelas.

Authorship

- 2019-03-22 Kris Coolsaet (BE) kris.coolsaet@ugent.be

License

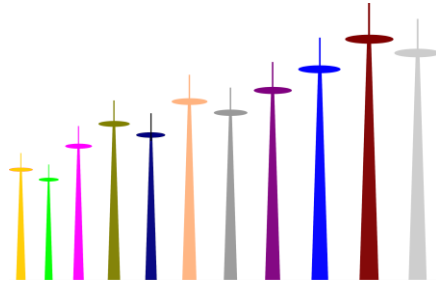
Copyright © 2019 Bebras – International Challenge on Informatics and Computational Thinking.
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



Menara Spesial

PENGGALANG
(SMP)
I-2019-CA-01

Lihatlah sederetan menara sebagai berikut:



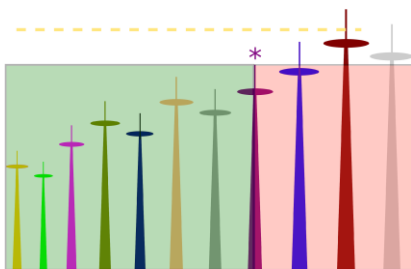
Sebuah menara dinyatakan spesial jika semua menara di kirinya lebih pendek, dan semua menara di kanannya lebih tinggi.

Tantangan: Ada berapa menara spesial pada gambar tersebut?

Jawaban:

Jawaban yang tepat adalah 3.

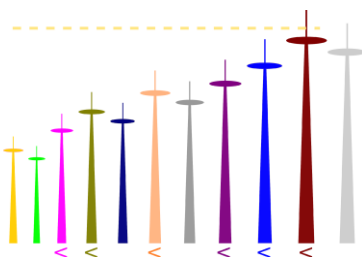
Sebuah menara adalah istimewa jika: semua ujung menara di sebelah kiri berada dalam persegi panjang berwarna (lihat gambar 1); dan jika semua ujung menara di kanan berada di luar persegi panjang berwarna (lihat gambar 2). Seperti yang ditunjukkan untuk menara kedelapan pada contoh di bawah ini.



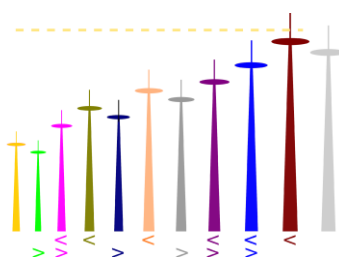
Gambar 1

Gambar 2

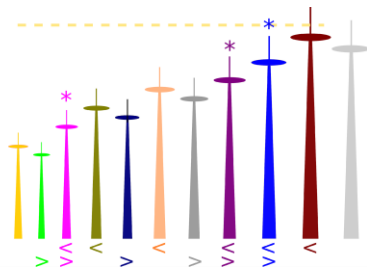
Salah satu cara yang mungkin untuk menemukan solusinya adalah bergerak melalui semua menara, satu per satu, dan menandai sebuah menara jika semua di sebelah kiri lebih pendek (lihat gambar 3). Kemudian buat lintasan kedua yang serupa, tandai menara jika semua ke kanan lebih tinggi (lihat gambar 4).



Gambar 3



Gambar 4



Semua menara yang ditandai dua kali (<,>) adalah menara khusus (*).

Ini Informatika!

Tugas ini berfokus pada urutan menara berdasarkan ketinggiannya. Secara umum, menempatkan benda ke dalam urutan disebut penyortiran yang merupakan salah satu masalah yang paling terkenal dan dipelajari dalam ilmu komputer. Ada banyak algoritma penyortiran yang berbeda. Algoritma quicksort adalah salah satu metode penyortiran yang paling terkenal dan sering digunakan karena rata-rata cenderung sangat cepat seperti yang ditunjukkan oleh namanya!

Algoritma quicksort bekerja sebagai berikut:

Elemen acak dari daftar dipilih. Elemen ini disebut "pivot". Semua elemen yang lebih kecil dari pivot dipindahkan ke sisi kiri pivot dan semua elemen yang lebih besar dari pivot dipindahkan ke sisi kanan pivot.

Sekarang ada dua daftar yang dihasilkan, satu ke kiri dan satu ke kanan dan proses yang sama diulang untuk keduanya secara rekursif. Setelah setiap langkah proses, daftar yang dihasilkan menjadi lebih kecil dan lebih kecil. Proses ini berakhir jika daftar yang dihasilkan hanya berisi 1 elemen, yang berarti bahwa seluruh daftar kemudian disortir. (Animasi quicksort)

Dalam tugas kami, menara khusus mewakili pivot setelah memindahkan elemen yang lebih kecil ke kiri dan elemen yang lebih besar ke kanan.

Authorship

- 2019-03-13 J.P. Prett (Canada), jpretti@uwaterloo.ca: Task Proposal
- 2019-05-15 Martina Palazzolo, martina.palazzolo.5@gmail.com, Bebras Workshop
- 2019-05-15 Allira Crowe (Australia), allira.crowe@csiro.au
- 2019-05-15 Wilfried Baumann, baumann@ocg.at, Bebras Workshop

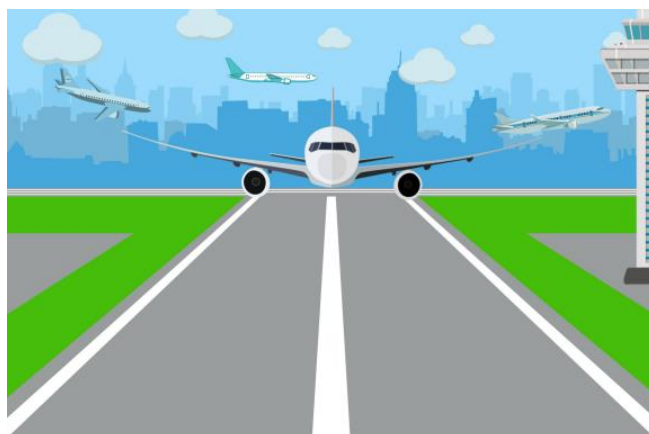
License

Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking. This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



Penjadwalan Penerbangan

PENGGALANG
(SMP)
I-2019-IN-03



Saat sebuah pesawat mendarat di bandara, pendaratan harus dilakukan di sebuah jalur khusus untuk menghindari tabrakan. Dua pesawat hanya boleh mendarat di jalur pendaratan jika selisih waktu kedatangannya berselisih 15 menit.

Misalnya Jika Penerbangan#1 mendarat pada Pukul 6:10 pagi, Penerbangan #2 mendarat pada Pukul 6:25 pagi and Penerbangan#3 mendarat pada 6:26 pagi, maka Penerbangan #1 dan Penerbangan#2 tidak dapat dijadwal untuk mendarat pada jalur pendaratan yang sama. Sedangkan Penerbangan#3 dan Penerbangan#1 dapat mendarat pada jalur yang sama, tapi tak boleh sama dengan jalur pendaratan Penerbangan#2.

Anda adalah pengatur pendaratan pesawat hari ini, dan jadwal pendaratan diberikan sebagai berikut:

Nomor penerbangan	Waktu
9W2400	7:00
9W1321	7:21
AI561	7:20
AI620	7:18
EK427	7:03
SG147	7:12

Tantangan: Berapa jumlah minimum jalur pendaratan yang dibutuhkan agar semua pesawat dapat mendarat sesuai jamnya?

Jawab dengan bilangan bulat antara 0 s.d 8.

Jawaban yang tepat adalah: 4.

Mari kita urutkan nomor penerbangan dalam urutan menaik berdasarkan waktu pendaratan.

1. 9W2400 7:00
2. EK427 7:03
3. SG147 7:12

- | | |
|-----------|------|
| 4. AI620 | 7:18 |
| 5. AI561 | 7:20 |
| 6. 9W1321 | 7:21 |

Penerbangan pertama yaitu 9W2400 mendarat pukul 7:00 sehingga kita tempatkan pada koridor 1. Dua penerbangan selanjutnya yaitu EK427 dan SG147 masih dalam rentang waktu 15 menit dari penerbangan pertama dan penerbangan ketiga masih dalam rentang waktu 15 menit dari penerbangan kedua, sehingga secara jelas ketiga penerbangan tersebut harus ditempatkan pada koridor yang terpisah. Sejauh ini kita membutuhkan tiga koridor.

Untuk setiap penerbangan berikutnya, kita mencoba untuk menempatkan pada koridor yang sudah ada, jika memungkinkan. Penerbangan keempat AI620 mendarat pukul 7:18 dimana waktunya lebih dari 15 menit sejak penerbangan pertama 9W2400 sehingga dapat kita tempatkan pada koridor yang sama dengan penerbangan 9W2400. Penerbangan kelima AI561 tidak bisa ditempatkan pada koridor 1 karena masih dalam rentang waktu 15 menit dari AI620, namun dapat ditempatkan pada koridor 2 karena lebih dari 15 menit dari EK427.

Penerbangan keenam 9W1321 mendarat pukul 7:21 sehingga kurang dari 15 menit dari penerbangan SG147 (7:12), AI620 (7:18), dan AI561 (7:20) yang sudah ditempatkan pada 3 koridor. Sehingga kita perlu menambahkan koridor baru untuk penerbangan ini.

Berikut adalah penempatan penerbangan pada koridor-koridor bandara:

Corridor 1: 9W2400 (7:00), AI620 (7:18)

Corridor 2: EK427 (7:03), AI561 (7:20)

Corridor 3: SG147 (7:12)

Corridor 4: 9W1321 (7:21)

Kita dapat menempatkan lebih dari 4 koridor, kurang dari 4 koridor maka dapat terjadi kecelakaan.

Ini Informatika!

Dalam persoalan di atas, Anda memiliki konflik (dua pesawat tidak bisa ditempatkan pada koridor yang sama jika harus mendarat dalam rentang waktu 15 menit satu sama lain) dan sumber daya (koridor) dan Anda harus menempatkan sumber daya untuk mengatasi konflik. Ada banyak persoalan dengan jenis ini. Contohnya, ketika Anda harus menentukan berapa banyak meja yang dibutuhkan dalam pesta pernikahan sehingga Anda dapat menempatkan tamu sedemikian rupa sehingga tidak ada tamu bermusuhan yang ditempatkan pada meja yang sama).

Ketika persoalan Anda memiliki banyak objek (pesawat terbang, tamu undangan, dll.), Anda ingin memecahkan persoalan dengan cepat. Informatika memiliki pendekatan khusus yang dikembangkan untuk memecahkan persoalan tersebut. Salah satunya adalah dengan menggunakan graf sebagai representasi. Sebuah graf memiliki simpul dan jalur yang menghubungkan simpul. Dalam kasus ini simpul melambangkan objek (penerbangan) dan jalur adalah konflik: jika dua objek berkonflik satu sama lain, maka simpul objek tersebut dihubungkan oleh sebuah jalur. Persoalan tersebut dapat dianggap sebagai persoalan pewarnaan graf: berapa jumlah minimum warna yang dibutuhkan untuk mewarnai setiap simpul pada graf dengan cara tertentu sehingga tidak ada simpul dengan warna yang sama terhubung dengan sebuah jalur.

Ketika diterapkan pada persoalan penjadwalan pesawat terbang, setiap warna akan merepresentasikan koridor. Pewarnaan graf menerapkan penjadwalan pesawat pada rute penerbangan, alokasi bandwidth pada stasiun radio, pencocokan pola, penjadwalan olahraga, desain

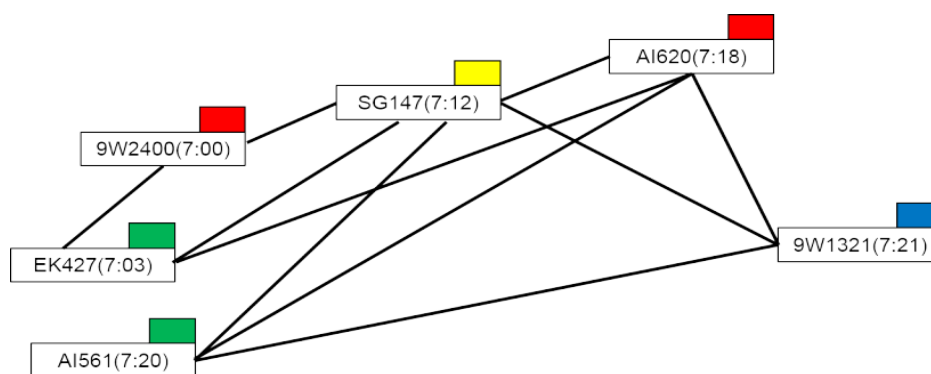
rencana penempatan kursi, penjadwalan ujian, penjadwalan taksi, dan bahkan pemecahan puzzle sudoku.

https://en.wikipedia.org/wiki/Graph_coloring

Ada banyak kemungkinan cara untuk menyelesaikan permasalahan ini, perbedaannya terletak pada struktur data (bagaimana kita merepresentasikan data) dan waktu yang kita butuhkan untuk sampai pada solusi. Salah satu metodenya adalah menggunakan algoritma greedy. Kata 'greedy' memiliki arti kita mencoba untuk menyortir semua objek dengan urutan menurun berdasarkan nilai yang kita berikan dengan penalaran. Pada kasus ini kita menggunakan waktu pendaratan sebagai nilai acuan (semakin awal pesawat mendarat, semakin tinggi nilainya). Algoritma greedy tidak selalu memberikan kita jawaban yang tepat, namun kebanyakan memberikan kita solusi yang sesuai.

https://en.wikipedia.org/wiki/Greedy_algorithm

Berikut adalah solusi menggunakan algoritma greedy dengan pewarnaan graf.



Masing-masing penerbangan memiliki relasi bahwa mereka berjarak 15 menit satu sama lain. Kita mulai dengan penerbangan 9W2400. Kita menghubungkan penerbangan tersebut dengan semua penerbangan yang berelasi, seperti penerbangan EK427 dan SG147 yang berjarak kurang dari 15 menit. Kita mengulangnya pada setiap penerbangan dengan membuat hubungan dengan penerbangan lain yang masih dalam rentang waktu 15 menit, jika sebuah koneksi belum dibuat.

Mari kita tempatkan koridor penerbangan, katakanlah warna merah, kepada penerbangan yang belum terhubung. 9W2400 ditempatkan pada koridor penerbangan 1 karena paling awal akan mendarat. Tidak semua penerbangan mendapatkan koridor penerbangan yang sama (warna merah). Koridor penerbangan berikutnya adalah koridor dua disimbolkan dengan warna hijau. Warna hijau diberikan ke EK427 dan penerbangan apapun yang tidak terhubung yaitu AI561. 9W1321 tidak bisa diberi warna hijau karena terhubung dengan AI561. Penempatan berikutnya koridor penerbangan ketiga (warna kuning) diberikan ke SG147 dan koridor penerbangan keempat (biru) ke 9W1321. Menjadi hal yang jelas bahwa kita membutuhkan tidak lebih dari empat koridor untuk persoalan di atas untuk memastikan semua penerbangan dapat mendarat dengan selamat.

Authorship

- 2019-02-14 Vipul Shah (India), v.shah@tcs.com
- 2019-02-14 Yogananda Jeppu (India), yvieppu@gmail.com

Task Proposal

Contributors:

- 2019-05-14 Špela Cerar (Slovenia), Ilya Kaysin (Russia): changed flight table (only list of flights, no arrivals or departures), made corrections to the task text, cleared explanation of solution

- 2019-05-15: Arnheidur Gudmundsdottir (Iceland), arnheidur@sky.is and Margot Phillipps (New Zealand), margot.phillipps@gmail.com . Made changes to the text in the problem statement. Added Air traffic Controller to the wording section.
- 2019-05-16: Alisher Ikramov (Uzbekistan), ikramov.alisher@list.ru; Humberto Sermeño (SV), humberto.sermenio@joventalento.edu.sv: made corrections to the task text, additions to the It's informatics section, added keyword.
- 2019-05-16: Margot Phillipps (New Zealand), margot.phillipps@gmail.com: changed age group difficulties, text corrections

License

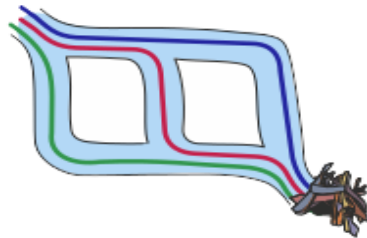
Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking. This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



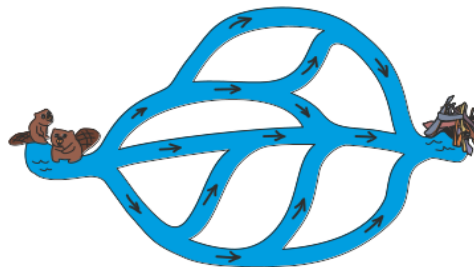
Jalan Pulang Berang-Berang

PENGGALANG
(SMP)
I-2019-LT-02

Para berang-berang ingin agar masing-masing mempunyai jalan pulang. Misalnya untuk kasus ini, ada 3 jalur yang berbeda, sehingga hanya 3 berang-berang yang dapat hidup pada sungai tersebut.



Ada sepasang berang-berang yang hidup di sisi sungai seperti ditunjukkan dalam gambar. Mereka ingin menambahkan anggota keluarga.



Tantangan: Berapa tambahan anggota keluarga maksimum yang dapat ditampung agar setiap bebras tetap mempunyai satu jalur jalan pulang?

Jawaban:

5.

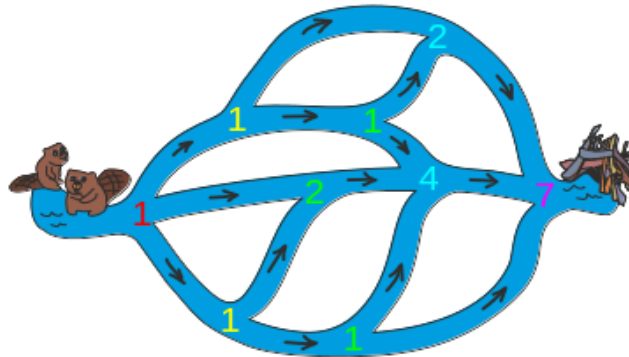
Ada 7 jalur berbeda yang mengarah ke rumah mereka, 2 di antaranya diperuntukkan bagi orang tua. Kita dapat menghitung jalur ini dengan dua cara berbeda.

Salah satu caranya adalah dengan mencari semua jalur dalam gambar dan menggambarinya dalam berbagai warna:



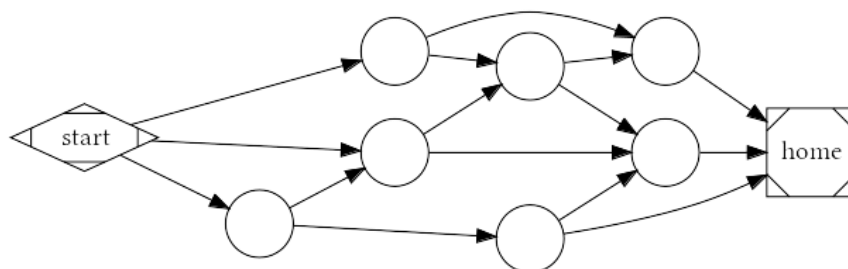
Cara kedua adalah dengan menggunakan metode pemrograman dinamis. Kita mencari semua persimpangan dalam gambar. Kita menghitung berapa banyak jalur yang mengarah ke setiap persimpangan dan menuliskan angka yang sesuai di sana. Untuk melakukannya, kita pergi dari kiri ke kanan dan meringkas angka-angka pada persimpangan yang terhubung langsung ke persimpangan ini dengan panah masuk.

Dalam hal ini, kita mulai dengan meletakkan 1 ke persimpangan awal (merah), karena hanya ada satu jalur dari posisi awal berang-berang. Kemudian kita menghitung angka untuk persimpangan kuning - mereka juga sama dengan 1, karena satu-satunya persimpangan yang mungkin sebelumnya adalah yang merah. Setelah itu kita menghitung persimpangan hijau: misalnya, untuk mendapatkan hijau 2 kita meringkas merah 1 dan kuning 1. Selanjutnya kita menghitung persimpangan biru, dan kemudian persimpangan akhir: kita mendapatkan 7 dengan menjumlahkan 2, 4 dan 1 dari persimpangan sebelumnya.



Ini Informatika!

Tugas yang berbeda memerlukan representasi data yang berbeda. Salah satu caranya adalah dengan menggunakan grafik. Sistem air dalam gambar dapat direpresentasikan sebagai grafik berarah. Tepi mewakili bagian sungai dan node mewakili persimpangan. Grafik untuk tugas kami terlihat seperti ini:



Graf berarah adalah struktur data yang dapat membantu kita untuk menentukan apakah kita dapat berpindah dari satu node ke node lainnya. Kita dapat menulis ukuran panjang dari setiap bagian sungai ke tepi yang sesuai dan mendapatkan graf berbobot. Kita menggunakan graf berbobot ini untuk menemukan jalur optimal antara posisi awal dan akhir.

<https://en.wikipedia.org/wiki/Graph>

Pendekatan sistematis di mana kita membangun solusi langkah demi langkah menggunakan langkah sebelumnya disebut pemrograman dinamis. Pendekatan ini banyak digunakan untuk menyelesaikan banyak masalah modern. Kita dapat menerapkan pemrograman dinamis hanya dalam kasus di mana setiap langkah selanjutnya hanya bergantung pada situasi saat ini dan bukan pada cara kita mencapainya.

https://en.wikipedia.org/wiki/Dynamic_programming

Authorship

- 2018-02-28 Valentina Dagienė, Lithuania: valentina.dagiene@mii.vu.lt;
Gabrielė Stupurienė, Lithuania: gabriele.stupuriene@mii.vu.lt

Contribution:

- 2019-05-15 Ilya Kaysin (Russia) ilya.s.kaysin@gmail.com, and Špela Cerar (Slovenia), spela.cerar@pef.uni-lj.si:
Changed explanation, It's Informatics section and question to make it more clear. Changed and made additional graphics for the task
- 2019-05-16: Alisher Ikramov (Uzbekistan), ikramov.alisher@list.ru; Humberto Sermeño (SV), humberto.sermeno@joventalento.edu.sv:
rewrote the task text, revised, additions to the "It's informatics" section.

License

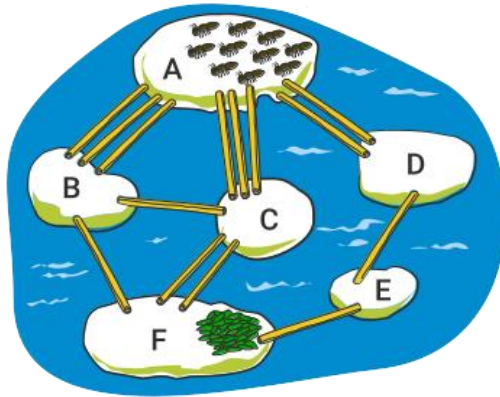
Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking.
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



Semut di Sarang

PENGGALANG
(SMP)
I-2019-LT-06

10 semut sedang berada di lokasi A dan mencoba untuk menuju ke F. Hanya 1 semut dapat lewat batang kayu setiap saat. Untuk pergi dari satu lokasi menuju lokasi lainnya dibutuhkan 1 menit.



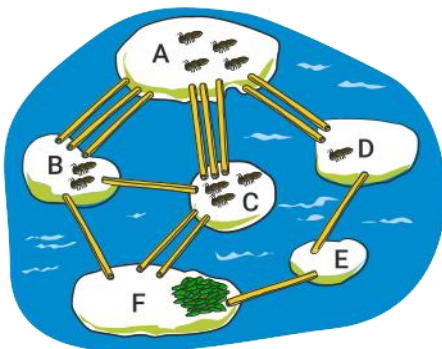
Tantangan: Berapa banyak semut dapat mencapai F dalam waktu 3 menit?
Jawab dengan bilangan bulat antara 0 s.d 10.

Jawaban:

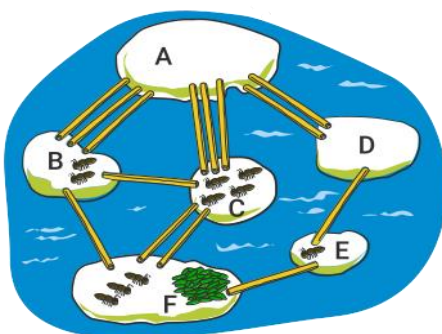
Jawaban yang tepat adalah 7.

Situasi yang mungkin terjadi setelah setiap menit disajikan pada gambar di bawah.

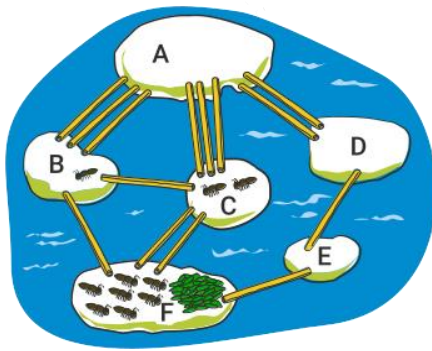
Setelah **satu** menit:



Setelah **dua** menit:



Setelah **tiga** menit:



Ini Informatika!

Tujuannya adalah optimasi alur berjalannya semut melalui jalur-jalur supaya sebanyak mungkin semut dapat sampai pada makanan dalam waktu 3 menit. Hal ini disebut sebagai persoalan optimasi.

Semut-semut tidak menyadari struktur dari jalur yang mereka lewati sehingga tidak akan bisa menemukan solusi terbaik. Namun sebagai pengamat yang dapat melihat struktur jalur, dapat menemukan strategi yang optimal. Dalam persoalan ini kita berasumsi bahwa semut-semut itu memahami struktur dari jalur, sehingga mereka dapat memilih jalur yang dilewati secara spesifik.

Graf adalah struktur data abstrak yang digunakan untuk memodelkan jalur dan banyak algoritma yang dapat digunakan untuk optimasi alur dalam keadaan yang spesifik. Secara alternatif tanpa menggunakan algoritma yang sudah ada, kita dapat menyatakan bahwa:

- tidak memungkinkan untuk mengirim lebih dari satu semut melalui batu D-E
- tidak memungkinkan untuk mengirim lebih dari dua semut melalui batu A-B
- jalur antara B-C tidak berkontribusi terhadap alur dan dapat diabaikan
- faktor yang membatasi alur adalah jalur B-F dan C-F

Pada kasus tersebut, pertimbangan-pertimbangan di atas memungkinkan untuk menemukan solusi yang optimal

https://en.wikipedia.org/wiki/Flow_network

https://en.wikipedia.org/wiki/Maximum_flow_problem

https://en.wikipedia.org/wiki/Optimization_problem

Authorship

- Valentina Dagienė, Lithuania: valentina.dagiene@mii.vu.lt

License

Copyright © 2019 Bebras – International Contest on Informatics and Computational Thinking.

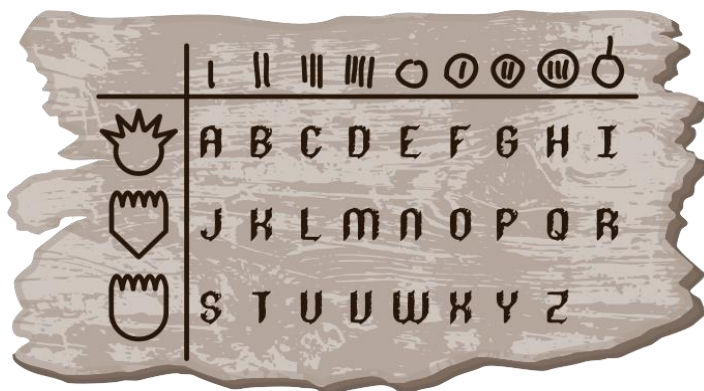
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). Visit: <https://creativecommons.org/licenses/by-sa/4.0/>



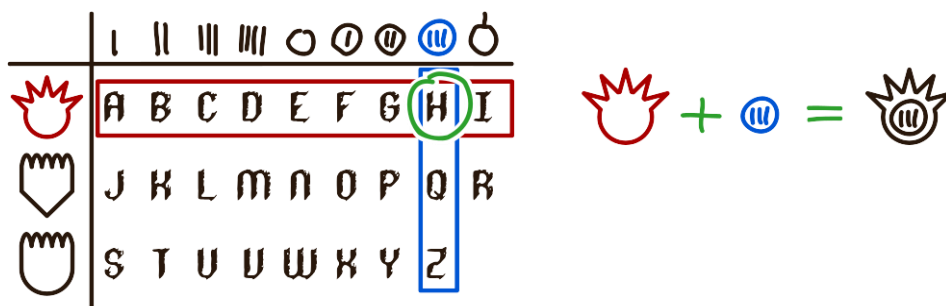
Pesan Nenek Moyang

PENGGALANG
(SMP)
I-2019-CH-10

Berang-berang si Pintar menemukan sebuah pohon tua yang berlubang. Ternyata, ada pesan rahasia yang diukir pada dinding lubang pohon tersebut. Ternyata, itu adalah kode yang ditulis nenek moyang saat masih hidup di celah tersebut.



Setelah berpikir cukup lama, ia menemukan kunci rahasianya: sebuah huruf abjad adalah kombinasi dari simbol pada judul baris dan kolomnya. Misalnya: huruf H dikode sebagai berikut:



Si Pintar menemukan pesan sebagai berikut:



Tantangan:

Pesan apa yang ditulis nenek moyang?

Pilihan Jawaban:

- a) LOVEWATER
- b) SLEEPDAYS
- c) LOVEMYSUN
- d) CAREFORME

Jawaban:

Jawaban yang benar adalah A) LOVEWATER.

Pada awalnya, kita memeriksa apakah panjang dari pesan memberikan petunjuk. Namun semua solusi memiliki panjang yang sama.



Dekode pada huruf pertama ditemukan simbol tersebut adalah "L". Hal ini memberitahu kita bahwa opsi jawaban B dan D tidak mungkin. Sehingga jawaban A dan C masih memungkinkan. adalah "L". Hal ini memberitahu kita bahwa opsi jawaban B dan D tidak mungkin. Sehingga jawaban A



dan C masih memungkinkan. adalah "W". Sehingga kita perlu mengecek apakah solusi A bernilai benar.

Untuk meyakinkan bahwa A adalah solusi yang tepat, kita dekode simbol lain yaitu simbol terakhir.



Simbol tersebut bernilai "R" sehingga kita dapat yakin bahwa A adalah jawaban yang tepat.

Ini Informatika!

Keamanan data adalah isu besar dalam masyarakat masa kini. Salah satu metode untuk melindungi data dari orang yang tidak berwenang adalah dengan penyandian rahasia. Kriptologi dimulai pada 3500 tahun yang lalu dan salah satu metode awal yang dikembangkan adalah mengganti setiap huruf dengan huruf yang lain. Pada contoh ini, simbol baru dibuat untuk mengganti huruf alfabet, dengan cara ini seseorang dapat dengan mudah mengingat cara penyandian.

Dalam persoalan ini jika tidak ada tabel yang mendeskripsikan pengkodean dan kita hanya diberikan simbol, membaca sandi simbol menjadi kata-kata akan menjadi lebih sulit. Kriptanalisis, seorang yang mencoba memecahkan kode, akan menggunakan teknik seperti analisis frekuensi dan pengenalan pola untuk menentukan pada arti dari simbol-simbol tersebut.

Authors dan Contributors (termasuk Graphics)

- Juraj Hromkovič, juraj.hromkovic@inf.ethz.ch, Switzerland.
- Regula Lacher: regula.lacher@inf.ethz.ch, Switzerland.
- Susanne Datzko: susanne@datzko.ch, Switzerland.
- Olivier Ens: olivier.ens@informatik-biber.ch, Switzerland.

License

Copyright © 2019 Bebras – International Contest on Informatics and Computer Fluency. This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.



<http://creativecommons.org/licenses/by-nc-sa/3.0/>