



EMPAT PERUNGGU DARI KAZAN, RUSSIA

Dalam suasana kemeriahan peringatan Hari Kemerdekaan RI yang ke 71, Tim Olimpiade Komputer Indonesia 2016 berhasil mempersembahkan empat medali perunggu dalam ajang International Olympiad in Informatics (IOI) ke-28 yang diselenggarakan di Kazan, Rusia pada tanggal 12 – 19 Agustus 2016. Mereka adalah :

- Kwee Lung Sin, SMA Karangturi Semarang
- Muhammad Yusuf Sholeh, SMA Al Irsyad Purwokerto
- Sergio Vieri, SMA Intan Permata Hati Surabaya
- Stacia Edina Johanna, SMA Petra 3 Surabaya

Dalam dua hari pertandingan yang dilakukan, Kwee Lung Sin berhasil mengumpulkan total nilai 306 (dari maksimal 600) dan menempati peringkat ke-93 dari 308 peserta, disusul oleh Stacia dengan nilai 289 yang menempati peringkat ke-114. Yusuf Sholeh berada di bawah Stacia, yaitu pada peringkat 139 dengan nilai 256. Sedangkan Sergio Vieri berada di posisi 141 dengan nilai total perolehan 251. Prestasi ini mengulang perolehan pada IOI ke 26 di Taiwan tahun 2014 lalu.

IOI ke-28 ini diikuti oleh 308 peserta dari 88 negara. Nilai tertinggi dalam kompetisi ini diraih oleh Ce Jin dari China yang meraih nilai 597 hasil dari dua hari pertandingan. Berbeda dengan tahun sebelumnya dimana absolute winner berhasil meraih nilai sempurna 600, hal ini menunjukkan bahwa soal yang diberikan dalam IOI 2016 kali ini memang lebih sulit. China berhasil keluar sebagai juara umum dengan memperoleh 4 medali Emas, sedangkan Rusia sebagai

tuan rumah berhasil meraih 3 medali Emas dan 1 medali Perak.

Dibanding dengan negara-negara Asia Tenggara seperti Malaysia dan Philipina (1 Perunggu), Indonesia masih memimpin dalam IOI 2016 kali ini, namun posisi tim Indonesia berada di belakang Singapura (1 Emas, 2 Perak dan 1 perunggu) dan Thailand (1 Perak dan 3 Perunggu). Oleh karena itu, usaha keras masih harus terus dilakukan agar perolehan medali emas di dalam IOI dapat terwujud pada tahun-tahun berikutnya.

Secara umum, penyelenggaraan IOI 2016 di Kazan, Rusia ini berjalan dengan baik dan lancar. Disamping acara utama IOI 2016, pada IOI kali ini diselenggarakan The 10th IOI Conference bersamaan dengan pelaksanaan IOI, dimana Tim Indonesia mempresentasikan dua paper dengan judul: (1) Reshaping Indonesian Students' Training for IOI (Inggriani Liem), dan (2) Understanding an Unsolvable Problem (Jonathan Irvin Gunawan). Presentasi tersebut mendapat apresiasi cukup positif dari negara-negara peserta lainnya, dengan harapan dari panitia agar Indonesia terus menulis paper untuk dipresentasikan pada IOI selanjutnya.



Agenda Kegiatan OSN 2017

- | | | | |
|---|---|---|--|
| 2 | Peserta tiba di Pekanbaru,
Check in Hotel Grand Elite | | |
| 3 | Upacara Pembukaan,
Penjelasan Peserta + Sesi Latihan,
Kampus Politeknik Caltex Riau | 8 | Peserta kembali
ke daerah masing-masing |
| 4 | Lomba Hari 1,
Kampus Politeknik Caltex Riau | 7 | Upacara Penutupan dan
Pembagian Medali |
| 5 | Lomba Hari 2
Gathering TOKI
Kampus Politeknik Caltex Riau | 6 | Wisata Pekanbaru dan sekitarnya
Pendidikan Karakter |

DAFTAR ISI

- 01** Empat Perunggu
dari Russia
- 02** Agenda OSN 2017
- 03** Daftar Isi
- 03** Selamat Datang
di Pekanbaru
- 04** Menuju IOI 2017
Tehran, Iran
- 06** Computational
Thinking
- 08** Bebras
Indonesia
- 10** Dari OSN
Ke Google
- 12** Style Lomba
Pemrograman
- 14** Pembahasan OSP
2017 (Algoritma)
- 16** Pembahasan OSP
2017 (Pemrograman)
- 19** Pembahasan OSN
2017 (Belanja Souvenir)
- 20** Hall of Fame
TOKI 2017

KONTRIBUTOR TOKINEWS 2017



Ogo baru OSN yang digunakan mulai OSN 2017 ini telah diresmikan pada penutupan OSN 2016 di Palembang oleh menteri Pendidikan dan Kebudayaan saat itu Anies Baswedan.

Logo OSN baru adalah gambar berbentuk rubik tiga sisi yang masing-masing sisinya bertuliskan huruf O, S dan N. Masing-masing sis-

memiliki warna berbeda sesuai jenjang yakni merah untuk sekolah dasar, biru untuk sekolah menengah pertama dan abu-abu untuk sekolah menengah atas. Logo ini menggantikan logo lama yang berbentuk tubuh seseorang sedang mengangkat tangan dengan bendera Indonesia di bagian kepala.

Logo baru OSN ini adalah logo karya Adhitya Dharmawan S yang berhasil memenangkan sayembara desain logo baru OSN yang dilaksanakan oleh Kemdikbud pada April 2017 lalu. (Sheila Rafiqah Uly, Xpresi)

Selamat datang di OSN 2017

Selamat datang di Pekanbaru, Riau

Riau salah satu propinsi di Indonesia yang terletak di bagian tengah pulau Sumatera. Provinsi ini terletak di bagian tengah pantai timur Pulau Sumatera, yaitu di sepanjang pesisir Selat Melaka. Riau saat ini merupakan salah satu provinsi terkaya di Indonesia, dan sumber dayanya didominasi oleh sumber alam, terutama minyak bumi, gas alam, karet dan kelapa sawit.



Kota Pekanbaru adalah ibu kota dan kota terbesar di provinsi Riau, Indonesia. Kota ini merupakan salah satu sentra ekonomi terbesar di bagian timur Pulau Sumatera, dan termasuk sebagai kota dengan tingkat pertumbuhan, migrasi dan urbanisasi yang tinggi. Kota ini berawal dari sebuah pasar (pekan) yang didirikan oleh para pedagang Minangkabau di tepi Sungai Siak. Kota Pekanbaru tumbuh pesat dengan berkembangnya industri terutama yang berkaitan dengan minyak bumi, serta pelaksanaan otonomi daerah.



Meskipun terkenal sebagai kota bisnis, akan tetapi Pekanbaru juga memiliki tempat wisata yang

jenis dapat dinikmati di sini, mulai dari wisata alam, wisata religi, wisata budaya dan juga tidak lupa wisata kuliner.

Di bidang pendidikan, Riau juga tergolong cukup maju, banyak siswa-siswi asal Riau yang memiliki prestasi baik di dalam olimpiade sains, baik tingkat nasional maupun Internasional. OSN 2017 ini merupakan OSN kedua yang dilaksanakan di Pekanbaru,

Untuk tahun 2017 ini kegiatan lomba OSN bidang Komputer/Informatika akan dilaksanakan di Kampus Politeknik Caltex Riau (PCR) yang berada di kawasan Rumbai, Pekanbaru.



sebelumnya Pekanbaru pernah menjadi tuan rumah pelaksanaan OSN ke 3 tahun 2004 lalu.

Selamat
Bertanding,
Jadilah yang
Terbaik ... !



TOKI 2017 BERSIAP

MENUJU TEHRAN, IRAN

oleh Fauzan Joko, Pembina TOKI

Tim Olimpiade Komputer Indonesia 2017 yang terdiri dari : (1) Sergio Vieri, (2) Kwee Lung Sin, (3) Steven Wijaya dan (4) Yehezkiel Reymond saat ini sedang mengikuti persiapan akhir untuk bertanding di ajang International Olympiad in Informatics ke 29 tahun 2019 di Tehran, Iran. 28 Juli s.d. 4 Agustus 2017.

Tim Olimpiade Komputer Indonesia 2017 merupakan sebuah tim yang dihasilkan melalui proses panjang dan bertahap. Banyak waktu, tenaga dan pikiran yang telah dicurahkan, tidak hanya dari peserta, namun juga para pembina, asisten, alumni TOKI dan juga dari pihak Kementerian Pendidikan dan Kebudayaan.

Perjalanan empat anggota tim tidaklah mudah, mulai dari tingkat sekolah, kabupaten/kota, tingkat propinsi dan juga tingkat Nasional dalam ajang Olimpiade Sains Nasional 206 lalu. Olimpiade Sains Nasional 2016 dilaksanakan di Kota Palembang, Sumatra Selatan. Untuk bidang Informatika/Komputer lomba dilaksanakan di Kampus MDP Palembang dan diikuti oleh 87 peserta dari 34 propinsi di Indonesia. Tiga puluh peserta yang mendapatkan medali dalam ajang ini (5 medali emas, 10 medali perak dan 15 medali perunggu) berhak mengikuti rangkaian pelatihan dan seleksi lebih lanjut.



Pembinaan Tahap 1.

Pelatnas 1 TOKI 2017 yang dilaksanakan di Institut Teknologi Bandung pada tanggal 26 September sampai dengan 19 Oktober 2017 ini telah berlangsung dengan lancar tanpa ada kendala. Selama kegiatan Pelatnas, kepada peserta diberikan materi-materi (ilmu baru) dan diwajibkan mengerjakan soal-soal latihan yang sudah disiapkan oleh Scientific Committee setiap harinya. Selain sesi latihan, terdapat pula Repeating yang dilaksanakan tiap akhir minggu, kontes Speed Practices yang mengasah kemampuan berpikir cepat untuk menyelesaikan problem, dan juga Kuis dan Simulasi yang dilakukan untuk mengukur kemampuan yang dimiliki peserta dan serta melatih kemampuan

stress management peserta apabila dihadapkan dengan kompetisi yang cukup besar seperti IOI.

Selain kegiatan yang bersifat memperkaya materi, kegiatan Pelatnas juga mengajak peserta untuk bersenang-senang serta berekreasi di Floating Market, Lembang dan Paris Van Java, Bandung. Selain itu, juga terdapat sesi sharing bersama Ikatan Alumni Tim Olimpiade Komputer Indonesia (IA-TOKI) di Wisma TOKI.



Pembinaan Tahap 2

Kegiatan Pelatnas tahap 2 Tim Olimpiade Komputer Indonesia (TOKI) 2017 dilaksanakan pada Kamis, 15 Februari 2017 hingga Sabtu, 11 Maret

2017 di Teknik Informatika Institut,Teknologi Sepuluh Nopember Surabaya. Kegiatan ini berlangsung selama tiga minggu dengan berbagai agenda kegiatan.

Kegiatan latihan dilakukan setiap hari dengan jadwal pagi pengerojan soal oleh peserta secara mandiri dan pada siang hari dilakukan pembahasan soal-soal bersama asisten.

Sepanjang rangkaian kegiatan, para peserta juga mengikuti kuis mingguan dan berbagai macam kontes, seperti : TOKI Open Contest 2017 dan Croatian Open Competition in Informatics sebagai kegiatan latihan. Diakhir masa pembinaan dilakukan kuis akhir dan simulasi yang merupakan tahap akhir pembinaan. Hasil evaluasi akan menjadi point penting untuk menentukan siapa saja yang berhak untuk lolos ke Pelatnas Tahap 3.



Materi yang diberikan antara lain Teknik Penyelesaian soal tingkat lanjut, ketrampilan implementasi program, simulasi penyelesaian soal dan soft skill dan motivation training. Di akhir pelaksanaan pelatnas dilakukan simulasi dan seleksi untuk menentukan 4 siswa terbaik untuk mewakili Indonesia di IOI 2017.

Pembinaan Tahap 4

Empat peserta yang lolos dari pembinaan tahap 4 ini selanjutnya akan mewakili tim Indonesia dalam ajang International Olympiad in Informatics ke 29 tahun 2017 yang akan dilaksanakan di Tehran, Iran. Namun sebelumnya para siswa akan mengikuti pembinaan tahap akhir di Fakultas Ilmu Komputer, Universitas Indonesia selama kurang lebih 2 minggu dengan lebih menekankan kepada latihan soal, diskusi dan simulasi soal-soal IOI.

Kita doakan bersama semoga semua jerih payah dan usaha yang telah di-lakukan ini terbayar dengan prestasi terbaik untuk mengharumkan nama bangsa Indo-nesia di dunia Internasional.



Pembinaan Tahap 3

Pelatnas 3 dilaksanakan selama 25 hari mulai dari tanggal 19 April s.d. 13 Mei 2017 di Kampus Departemen Ilmu Komputer, FMIPA, Institut Pertanian Bogor. Kegiatan utama selama pelatnas adalah Pengajaran di Keals, Latihan Praktek, Latihan mandiri, kuis, simulasi, pembahasan soal dan kegiatan non teknis untuk tujuan kebersamaan dan semangat para peserta.

Computational Thinking

oleh Inggriani Liem, Pembina TOKI

Computatio

Kemampuan berpikir kreatif, kritis dan komunikasi serta kolaborasi adalah kemampuan yang paling penting dalam (21st century learning) pembelajaran di abad kedua-puluhan-satu, di antara kemampuan-kemampuan lainnya seperti membaca, matematik, sains. Siswa zaman sekarang perlu untuk mengembangkan keterampilan berpikir, 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 dapat secara efektif dilaksanakan oleh sebuah agen pemroses informasi ("komputer", robot, atau manusia). CT adalah sebuah proses pemecahan persoalan dengan tahapan:

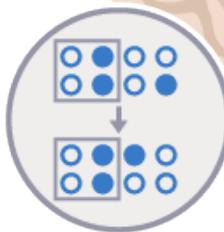
- Memformulasikan persoalan sedemikian rupa sehingga dapat menggunakan komputer sebagai alat bantu.
- Mengorganisasi dan menganalisis data secara logis.
- Merepresentasi data melalui abstraksi seperti model dan simulasi
- Mengotomasi solusi melalui berpikir algoritmik (langkah-langkah yang terurut)
- Mengidentifikasi, menganalisis, dan mengimplementasi solusi yang mungkin dengan tujuan langkah dan sumberdaya yang efisien dan efektif
- Melakukan generalisasi dan mentransfer proses pemecahan persoalan ke berbagai jenis persoalan

CT adalah sebuah cara untuk memecahkan persoalan, merancang sistem, memahami perilaku manusia yang melandasi konsep informatika. Dalam dunia saat ini, CT harus menjadi dasar bagaimana seseorang berpikir dan memahami dunia. CT berarti menciptakan dan menggunakan beberapa level 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.

Decomposition



Pattern recognition



nal thinking



Di negara maju, "Computer Science" (yang di Indonesia diterjemahkan sebagai "Informatika") sudah mulai diajarkan di tingkat SD, mengacu ke kurikulum yang disusun oleh persatuan guru-guru, asosiasi profesi informatika. Perancang model kurikulum juga menyediakan sejumlah sumber daya belajar untuk dimanfaatkan para guru. Salah satu model kurikulum "Computer Science" dikembangkan oleh ACM dan asosiasi guru [<https://csta.acm.org/Curriculum/sub/CurrResources.html>] terdiri atas 4 tingkatan sebagai berikut: Dasar Ilmu Komputer (mulai K-8/SMP), Ilmu Komputer di dunia modern (untuk K9 dan K-10), Analisis dan Desain menggunakan Ilmu Komputer, Topik Ilmu Komputer (K-11, K-12).

Di Inggris, sejak tahun 2014 Computer Science pun mulai digagas sebagai salah satu cabang ilmu melengkapi STEM (Science, Tehcnology, Engineering, Mathematics) menjadi STEM-Computing (STEM-C) yang menjadi mata pelajaran wajib sampai saat ini.

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

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. Kemampuan TIK mengarah ke penggunaan teknologi dan perangkat/gadget, sedangkan Informatika mengarah ke keilmuan komputasinya. Penggunaan TIK yang dimaksud bukan hanya ketrampilan menggunakan gadget dan aplikasinya, tetapi juga kemampuan untuk menggunakan dan memanfaatkan konten dengan bijak.

Bebras Computational Thinking Challenge

oleh Inggriani Liem, Pembina TOKI

Apa itu Bebras Computational Thinking Challenge ?

"Bebras" dalam bahasa Lituania berarti berang-berang, binatang pengerat yang cerdas serta banyak akal sehingga dapat melakukan banyak hal, misalnya membendung sungai untuk menangkap ikan dengan kayu yang dipatahkannya dengan giginya yang kuat.

Bebras challenge (semula adalah algoritmic challenge kemudian menjadi computational thinking challenge), disinisasi oleh Prof. Valentina Dagiene dari Lithuania sejak tahun 2004, adalah kompetisi yang diadakan tahunan bagi siswa berumur 5 s.d. 18 tahun dan sudah diikuti oleh sekitar 1.3 juta siswa yang berasal dari 50 negara. Komunitas Bebras sebagian besar adalah para pembina IOI seperti halnya Indonesia, adalah sekumpulan akademisi yang peduli ke pendidikan dasar dan menengah.

Bebras mengikuti perkembangan CT ini, lewat "challenge" atau tantangan yang diberikan untuk problem solving terkait informatika untuk kehidupan sehari-hari, yang disajikan secara menarik, menyenangkan, dan lucu. Lewat Tantangan Bebras, siswa diajak "membangun" ketrampilan 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.

Tantangan bebras diselenggarakan pada saat hampir bersamaan di seluruh dunia, pada minggu Bebras, yang ditetapkan pada minggu pertama bulan November.

Tujuan Tantangan Bebras :

- Memotivasi siswa agar mulai tertarik ke topik-topik informatika dan memecahkan persoalan dengan menggunakan ilmu 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 drpd sekedar ke komputer/alatnya dan TIK



TOKI secara resmi dipercaya untuk menyelenggarakan Bebras di Indonesia



Salah seorang siswa mengikuti Bebras Challenge

Bagaimana dengan Indonesia ?

(<http://bebras.or.id>)

Bebras dikelola oleh pembina Pusat/Nasional TOKI. Indonesia mulai bergabung ke komunitas internasional bebras pada bulan Mei 2016, kemudian mengadakan Tantangan Bebras dalam bahasa Indonesia pada bulan November 2016. Tantangan Bebras Indonesia telah diselenggarakan secara online di gelar di 12 kota dengan 1553 peserta yang terdiri dari 193 peserta kategori Siaga (SD), 198 peserta kategori Penggalang (SMP), dan 968 peserta kategori Penegak (SMA/SMK/MAN) yang berasal dari 125 Sekolah (23 SD, 27 SMP, 75 SMA). Penyelenggaraan Tantangan Bebras dikoordinasi oleh Perguruan Tinggi yang menjadi Mitra bebras Indonesia, dan diselenggarakan di Perguruan Tinggi Koordinator atau di sekolah. Peserta ada yang menggunakan komputer, tablet, bahkan handphone.

Setelah mendapat kesempatan menjadi observer pada workshop Internasional Bebras di Bodrum Turki pada tahun 2016, pada bulan April 2017 Indonesia mengajukan diri untuk menjadi **National Bebras Organizer** (NBO). Permohonan tersebut diproses ber-

sama 15 negara pemohon lainnya. Pada workshop bebras internasional 2017 yang diselenggarakan di Brescia-Italia tanggal 29 Mei s.d. 2 Juni 2017. Bebras Board telah memutuskan 11 negara baru termasuk Indonesia yang dapat diterima sebagai NBO mulai thn 2017. Dengan demikian anggota resmi komunitas Bebras saat ini sudah menjadi **52 negara** (<http://bebras.org>)

Dengan meningkatnya status Indonesia dari observer menjadi NBO, Bebras Indonesia mempunyai kewajiban mengusulkan soal, mengirimkan delegasi untuk hadir di workshop bebras Internasional tahun 2018, dan menyelenggarakan tantangan bebras 2017 sesuai jadwal yang ditetapkan. Jadwal tantangan Bebras 2017 adalah pada periode 16-19 November 2017. Sedangkan workshop bebras internasional tahun 2018 akan diadakan di Cyprus pada bulan Mei 2018 yang akan datang.

Bagaimana Untuk Berpartisipasi pada Tantangan Bebras 2017 ?

Pembina Bebras Indonesia bekerja sama dengan Perguruan Tinggi mitra, serta dukungan supporter. 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 workshop/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>

Untuk Latihan di arena Bebras Indonesia

Siapapun dapat 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 (minimal 8 karakter terdiri dari angka, huruf kecil, huruf besar, dan karakter khusus), anda menekan "Create New Account", kemudian anda harus membuka email untuk konfirmasi. Email anda harus email yang valid dan aktif.
3. Buka email anda.
4. Klik link yang diberikan oleh Olympia ke email anda.
5. Anda melakukan "Enroll " ke Bebras Challenge.



Referensi

- | | |
|------------------------|---|
| 21st century learning | : http://www.p2i.org/ |
| Computational Thinking | : https://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf |
| Untuk STEMC | : https://en.wikipedia.org/wiki/Computational_thinking |
| Constructionism | : https://www.nsf.gov/funding/pgm_summ.jsp?pgms_id=503582 |
| Kurikulum CS | : https://en.wikipedia.org/wiki/Constructionism_%28learning_theory%29 |
| | : https://csta.acm.org/Curriculum/sub/CurrResources.html |

DARI OSN KE Google

oleh Veni Johanna, Alumni TOKI

Di liburan musim panas tahun 2016, Sharon Lynn dan Teresa Pranyoto mencapai impian banyak pembela-jar Ilmu Komputer di seluruh dunia: bekerja di Google. Sharon, mahasiswi National University of Singapore (NUS), magang di tim Voice Search di kantor pusat Google di Mountain View, sementara Teresa terbang ke Google Sydney. Saat itu, Teresa baru menyelesaikan tahun ketiga studinya di Fakultas Ilmu Komputer di Universitas indonesia.

Kata ‘magang’ terkadang diasosiasikan dengan pekerjaan-pekerjaan ringan yang tidak edukasional, seperti membuat fotokopi, menulis catatan, atau membuatkan kopi. Sharon, Teresa, dan banyak pegawai magang Google lainnya diberi kepercayaan besar dalam mengerjakan projek-projek yang juga penting untuk perusahaan. Ditambah dengan adanya para mentor yang membimbing mereka, pegawai magang di Google dan banyak perusahaan teknologi lainnya bisa cepat mengembangkan kemampuan dan skills mereka.

Bekerja dan belajar bersama orang-orang terbaik dunia di salah satu perusahaan teknologi paling tenar di dunia adalah sesuatu yang telah lama diinginkan kedua mahasiswa Indonesia ini. “Saya sangat senang dan bersemangat, tapi juga tidak percaya kalau saya benar-benar diterima di Google,” kata Sharon, mengingat saat di mana ia mendapat surat penerimaan dari Google. “Tidak percaya, karena saat mendaftar saya merasa pesimistik. Senior-senior yang bekerja ke Silicon Valley terasa sebagai orang-orang yang jago banget, sementara saya merasa biasa saja. Tapi kerja keras saya betul-betul ada hasilnya.”



Bagi Sharon dan Teresa, pengalaman asyik bekerja di Google semakin menetapkan kecintaan mereka akan bidang ilmu komputer. Untuk Teresa, kecintaannya bermula dari mobile development and game development, yang ia temukan di tingkat pertama kuliah di Universitas Indonesia. “Dari program, kita bisa membuat sesuatu yang menyenangkan... Apalagi aku suka main game. Aku senang bisa menggunakan apa yang aku buat.” Sementara itu, di kuliah, Sharon menyadari bahwa aplikasi ilmu komputer tidak hanya sekadar membuat web atau app saja, tapi juga di banyak bidang-bidang lain, seperti biologi atau finansial. Potensial ilmu ini sangat besar - untuk orang yang mau belajar, kemungkinan yang ada tak terbatas.

Tahun ini, keduanya akan lulus kuliah dan akan kembali ke Google, kali ini sebagai pegawai tetap: Sharon di Google Singapura,

ra, dan Teresa di Google Sydney, Australia. Namun masa depan ini sama sekali tidak terlintas di pikiran mereka lima tahun lalu, ketika mereka gagal meraih medali di Olimpiade Sains Nasional bidang komputer.



Kegagalan ini meluluhkan hati mereka. Setelah OSN, mereka berdua bahkan sempat berpikir untuk kuliah di jurusan lain: Sharon di Ilmu Kedokteran, Teresa di Teknik Metalurgi. Mereka sempat merasa takut bahwa mereka tidak cocok untuk mendalami Ilmu Komputer, setelah merasa bodoh dan tertinggal dibandingkan orang-orang lain yang mereka temui di OSN.



"Perasaan itu sebenarnya baik juga; kalau saya nggak pernah merasa bodoh, artinya saya terlalu santai dan cepat puas," kata Sharon, mengingat perasaan mindernya saat di OSN. Di sisi lain, dengan mengetahui banyak orang melalui OSN, mereka juga termotivasi dan terinspirasi melihat prestasi partisipan lain atau alumni Tim Olimpiade Komputer Indonesia (TOKI). Misalnya, alumni-alumni TOKI telah memulai atau memimpin perusahaan-perusahaan teknologi startup di Indonesia, seperti Traveloka, Agate Studio, Sirclo, dan Urbanindo. Di luar negeri, alumni TOKI tersebar di perusahaan-perusahaan seperti Facebook, Google dan Quora.

Bagi Teresa, mantan wakil provinsi Riau, OSN adalah 'pembuka jalur' yang memotivasi dirinya untuk menjelajahi aplikasi-aplikasi ilmu komputer yang lain. Teknik-teknik yang ia pelajari juga berguna di kuliah, magang, atau proses wawancara kerja, seperti penggunaan dynamic programming dalam investigasi biologis tentang DNA. Buat Sharon, OSN berperan besar

dalam mendorong dirinya keluar dari zona nyaman. Pengalaman dalam belajar hal-hal baru yang sulit secara intensif juga membangun kebiasaan belajar dan berlatih yang kuat, yang terus dibawa ke universitas dan ke tempat kerja.

Sharon dan Teresa menunjukkan bahwa kegagalan di OSN bukanlah akhir, namun pembelajaran dan awal karir seru di bidang ilmu komputer. Yuk, kamu juga bisa!

Gaya (Style) Kontes Pemrograman

oleh Jonathan Irvin Gunawan, Alumni TOKI



<http://www.techrise.me/blog>

Selain Olimpiade Sains Nasional (OSN), terdapat banyak kontes pemrograman lainnya: International Olympiad in Informatics (IOI), Compfest oleh Universitas Indonesia, ACM International Collegiate Programming Contest (ICPC), TOKI Open Contest, dan lain-lain. Namun, berdasarkan stylenya sebenarnya kontes-kontes ini hanya terbagi menjadi dua kategori besar.

Kategori pertama adalah kontes bertipe IOI-style. Seperti namanya, IOI tentunya adalah salah satu kontes yang masuk ke kategori ini. Sebagai bagian dari persiapan IOI, OSN dan Pemusatan Latihan Nasional (Pelatnas) pun juga mengikuti style ini.

Kategori kedua adalah kontes bertipe ICPC-style. Style ini lebih sering digunakan untuk lomba-lomba online dan lomba-lomba pada jenjang mahasiswa, seperti ACM ICPC, Gemastik, TOKI Open Contest, dan lain lain.

Apa perbedaan dari kedua style kontes ini? Mere-

ka sangat berbeda. Umumnya, kontes bertipe ICPC-style memiliki lebih banyak soal daripada kontes bertipe IOI-style. Dalam waktu yang sama (lima jam), peserta IOI diberikan 3 soal namun peserta ICPC diberikan 10 sampai 13 soal. Peserta TOKI Open Contest pun diberikan 4 sampai 5 soal untuk diselesaikan hanya dalam 2 jam.

Proses penilaian dari kedua kontes ini pun juga berbeda. Umumnya, kontes bertipe IOI-style memiliki beberapa subsoal (subtask) dengan nilai yang berbeda-beda pada setiap soalnya, yang mana nilai dari setiap subsoal akan diperoleh jika peserta berhasil menyelesaikan seluruh kasus uji pada subsoal tersebut. Peringkat peserta ditentukan oleh jumlah nilai yang diperoleh pada setiap soal. Namun, kontes bertipe ICPC-style tidak memiliki subsoal pada setiap soalnya. Sehingga, peserta tidak akan mendapatkan nilai parsial untuk sebuah soal. Peserta akan menyelesaikan soal tersebut secara penuh atau tidak sama sekali.



Karena tidak adanya subsoal, sering sekali terdapat beberapa peserta dengan nilai yang sama. Karenanya, jika terdapat dua peserta dengan nilai yang sama, peringkat peserta ditentukan oleh total waktu yang dihabiskan untuk menyelesaikan soal-soal tersebut. Karenanya, banyak hal yang harus dipersiapkan untuk berpindah dari mengikuti kontes bertipe IOI-style menuju kontes bertipe ICPC-style. Bagi Anda yang sudah terbiasa mengikuti kontes bertipe IOI-style dan akan melanjutkan ke jenjang mahasiswa, ada baiknya untuk mempersiapkan diri untuk mengikuti kontes bertipe ICPC-style. Apa saja yang harus dipersiapkan? Salah satu hal penting adalah dengan melatih kecepatan coding Anda. Karena banyaknya soal yang relatif sedikit, biasanya Anda tidak memerlukan kecepatan coding yang tinggi untuk mermenangkan kontes bertipe IOI-style. Soal-soal pada kontes ini biasanya difokuskan pada observasi-observasi algoritma yang dalam. Namun pada kontes bertipe ICPC-style, Anda mungkin saja memerlukan kecepatan coding yang tinggi karena banyaknya soal yang diberikan. Selain itu, karena total waktu mempengaruhi peringkat, peserta yang memiliki kecepatan coding yang tinggi akan memiliki keuntungan yang cukup signifikan.

Hal penting lainnya adalah untuk mempersiapkan kemungkinan materi yang mungkin diujikan. Kontes IOI memiliki silabus yang berisi materi-materi yang mungkin diujikan. Biasanya kontes ini memfokuskan pada observasi peserta dan tidak akan mengujikan hal-hal yang membutuhkan teknik mendalam, seperti Fast Fourier Transform atau struktur

data kompleks seperti Link-Cut Tree. Karenanya, peserta yang belum mempelajari hal-hal tersebut tidak akan dirugikan. Namun, kontes bertipe ICPC-style biasanya tidak memiliki silabus. Materi apapun dapat diujikan oleh panitia. Karenanya, langkah baiknya bagi peserta untuk mempelajari sebanyak mungkin materi sebelum menghadapi kontes bertipe ICPC-style.

Kontes-kontes onsite pada jenjang mahasiswa yang bertipe ICPC-style, termasuk ICPC itu sendiri biasanya merupakan kontes tim. Untuk itu, juga merupakan hal yang penting untuk memenangkan kerja tim (teamwork) ketika menghadapi kontes ICPC atau kontes serupa.

Namun, hal terpenting untuk memenangkan kedua tipe kontes ini sebenarnya sama, yaitu banyaknya latihan soal, pembelajaran materi, dan kemampuan penyelesaian masalah (problem solving). Jika Anda sudah sering memenangkan kontes bertipe IOI-style, dengan beberapa latihan khusus untuk menghadapi kontes bertipe ICPC-style, Anda juga akan dapat memenangkan kontes bertipe ICPC-style, termasuk ICPC itu sendiri.

Ingatlah bahwa OSN bukan satu-satunya ajang kontes pemrograman. Masih banyak juga kontes lainnya yang dapat Anda manangkan diluar sana, termasuk pada jenjang mahasiswa. Selamat berjuang!



Pembahasan OSP Informatika 2017

Soal Algoritma

oleh Anthony Ng, Alumni TOKI

41

Diberikan fungsi seperti di bawah ini. Berapakah nilai yang dikembalikan dari pemanggilan fungsi `cariNilai(2017,1017)`?

```
function cariNilai(n, x : integer): longint;
var
    i : integer;
    n_now, x_now : integer;
    ans : longint;
begin
    ans := 0;
    n_now := n;
    x_now := x;
    while (n_now > 0) do begin
        ans := ans + x_now;
        x_now := x_now - 1;
        n_now := n_now - 1;
    end;
    cariNilai := ans;
end;
```

Untuk mempermudah pengerajan soal ini dapat digunakan rumus deret aritmetika.

$$S(n) = \frac{n}{2} (2a + (n-1)b)$$

Berikut rumus hasil dari deret aritmetika dengan nilai awal a , selisih b , dan banyaknya elemen n :

Pada soal ini, $a = 1017$, $b = -1$, $n = 2017$, dengan mensubstitusikan nilai-nilai tersebut ke dalam rumus $S(n)$ di atas, diperoleh jawaban akhir **18153**.

42

Diberikan fungsi seperti di bawah ini. Berapakah nilai dari `f(8, 9, 4, 5, 6, 10000)`?

```
function f(a, b, pa, pb, pab, n: longint): longint;
var
    total: longint;
    i: longint;
begin
    i := 0;
    total := 0;
    while (total < n) do
    begin
        i := i + 1;
        if (i mod a = 0) and (i mod b = 0) then
            total := total + pab
        else if i mod a = 0 then
            total := total + pa
        else if i mod b = 0 then
            total := total + pb;
    end;
    f := i;
end;
```

Fungsi **f(a, b, pa, pb, pab, n)** akan mencari nilai **i** terkecil sehingga **total >= n**. Untuk suatu nilai **i**, kita dapat menghitung nilai **total** sebagai berikut. Perhatikan bahwa ekspresi **if (i mod a = 0) and (i mod b = 0)** akan terjadi setiap KPK(**a, b**) sekali. Maka:

$$\begin{aligned} \text{total} &= (i / \text{KPK}(a, b)) * \text{pab} \\ &\quad + (i / a - i / \text{KPK}(a, b)) * \text{pa} \\ &\quad + (i / b - i / \text{KPK}(a, b)) * \text{pb} \end{aligned}$$

Jika disederhanakan akan menjadi:

$$\begin{aligned} \text{total} &= (i / \text{KPK}(a, b)) (\text{pab} - \text{pa} - \text{pb}) \\ &\quad + (i / a) * \text{pa} + (i / b) * \text{pb} \end{aligned}$$

Substitusikan nilai pada parameter:

$$\begin{aligned} \text{total} &= (i / 72) (6 - 4 - 5) \\ &\quad + (i / 8) * 4 + (i / 9) * 5 = 73 / 72 * i \end{aligned}$$

Karena kita ingin agar **total >= n**, maka substitusikan total dengan 10000 (dari nilai **n** pada parameter), diperoleh:

$$\begin{aligned} 10000 &= 73 / 72 * i \\ i &= 9863.0137 \end{aligned}$$

Karena nilai **i** yang diperoleh lebih dari 9863, sedangkan yang diinginkan adalah bilangan bulat, maka bilangan bulat terkecil terkecil **i** yang menyebabkan **total >= n** adalah **9864**.

44

Diberikan program di bawah ini. Apa output dari program tersebut apabila diberikan input (secara berurutan) yaitu 1000 dan 2?

```
function apaini(x:integer): integer;
var
  ans, i : integer;
begin
  ans := 0;
  i := 1;
  while (i <= x) do begin
    if (x mod i = 0) then
      ans := ans + 1;
    i := i + 1;
  end;
  apaini := ans;
end;
```

```
{Program Utama}
var
  i : integer;
  N : integer;
  x : integer;
  ans : integer;
begin
  readln(N, x);
  ans := 0;
  for i := 1 to N do begin
    if (apaini(i) mod x = 0) then
      ans := ans + 1;
  end;
  writeln(ans);
end.
```

Fungsi **apaini(x)** mengembalikan banyaknya faktor positif berbeda dari **x**. Dalam teori bilangan, terdapat rumus menghitung banyaknya faktor positif berbeda dari **N**:

Jika $N = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$, dengan semua p_i merupakan faktor-faktor prima dari N dan a_i adalah banyaknya kemunculan faktor prima p_i dalam N , maka banyaknya faktor positif berbeda dari N adalah

$$F(N) = (a_1 + 1)(a_2 + 1) \dots (a_k + 1)$$

Sebagai contoh jika $N = 72 = 2^3 3^2$, maka banyaknya faktor positif berbeda dari 72 adalah $F(72) = (3+1)(2+1) = 12$.

Program pada soal nomor 44 menerima dua buah nilai **N=1000** dan **x=2**, kemudian mencari banyaknya bilangan asli yang tidak lebih dari

1000 yang jika diaplikasikan fungsi **apaini()** dan dimodulo dengan 2 akan menghasilkan nilai 0 (dengan kata lain **apaini(i)** genap). Berdasarkan rumus $F(N)$ di atas dapat diperhatikan bahwa $F(N)$ akan bernilai genap jika terdapat **setidaknya satu** buah nilai a_j yang ganjil.

Untuk mempermudah, kita dapat menghitung komplemennya, yaitu banyaknya bilangan asli tidak lebih dari 1000 yang jika diaplikasikan dengan fungsi F hasilnya ganjil, atau dengan kata lain **hanya ada (semua) a_j yang genap**. Bilangan-bilangan yang memenuhi sifat tersebut hanyalah bilangan kuadrat (1, 4, 9, 16, ...). Bilangan kuadrat terbesar tidak lebih dari 1000 adalah $31^2 = 961$, maka jawaban untuk soal ini adalah komplemennya yakni $1000 - 31 = 969$.



Pembahasan OSP Informatika 2017

Soal Pemrograman

oleh Ammar Fathin Sabili, Alumni TOKI

► Grid Berpolo

1	1	1	1	1	1	1	1	1
1	2	2	2	2	2	2	2	1
1	2	3	3	3	3	2	1	
1	2	3	4	4	3	2	1	
1	2	3	3	3	3	2	1	
1	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1

Format Masukan:

Sebuah baris yang berisi 4 bilangan bulat berturut-turut yakni N, M, X, dan Y.

Batasan:

$$1 \leq N, M \leq 10^9$$

$$1 \leq X \leq N$$

$$1 \leq Y \leq M$$

Format Keluaran:

Sebuah baris yang berisi sebuah bilangan bulat yang menandakan nomor berapa yang ada di baris ke-X dan kolom ke-Y dari grid berukuran N baris dan M kolom.

Diberikan grid berukuran N baris dan M kolom. Dari baris teratas hingga terbawah dinomori dengan baris 1 hingga baris N, dan dari kolom terkiri hingga terkanan dinomori dengan kolom 1 hingga kolom M.

Setiap petak pada grid diberi nomor dengan aturan pola berikut. Dimulai dari lapisan terluar grid diberi nomor 1, lapisan berikutnya

yang lebih dalam diberi nomor 2, lapisan berikutnya diberi nomor 3, dan seterusnya. Tabel di samping adalah contoh grid berukuran 7 baris dan 8 kolom yang telah diberi nomor.

Diberikan nilai N, M, X, dan Y. Anda diminta untuk membuat sebuah program yang menentukan nomor berapa yang ada di baris ke-X dan kolom ke-Y dari grid berukuran N baris dan M kolom.

Contoh Masukan	Contoh Keluaran
7 8 1 1	1
7 8 3 6	3
7 8 6 5	2
7 8 4 5	4
7 8 7 8	1
999 999 500 500	500

Salah satu solusi naif yang dapat dilakukan adalah dengan membuat sebuah array 2 dimensi kosong berukuran N x M, kemudian mengisi array tersebut dengan pola yang diberikan pada soal, dan terakhir mencetak isi dari array pada indeks [X,Y]. Namun, solusi naif ini tidak akan bekerja dengan batasan yang diberikan karena akan membutuhkan memori yang sangat besar. (Perhatikan bahwa dengan memori yang rasional, ukuran array maksimal biasanya hanya dapat mencapai puluhan juta indeks saja.) Bahkan, algoritma untuk mengisi array-nya pun tidak begitu sederhana.

Pendekatan lain yang dapat dilakukan adalah dengan berpikir terbalik. Alih-alih mencari tahu angka berapa yang berada pada indeks [X, Y], cobalah berpikir suatu angka Z berada pada indeks yang mana saja?

Dengan berpikir terbalik seperti ini, kita dapat menemukan fakta bahwa:

- Setiap angka 1 berada pada baris paling atas pertama, baris paling bawah pertama, kolom paling kiri pertama, dan kolom paling kanan pertama;
- Setiap angka 2 berada pada baris paling atas ke-2, baris paling bawah ke-2, kolom paling kiri ke-2, dan kolom paling kanan ke-2;
- Setiap angka 3 berada pada baris paling atas ke-3, baris paling bawah ke-3, kolom paling kiri ke-3, dan kolom paling kanan ke-3;
- dst.

Kesimpulan yang dapat diambil adalah dengan mencari tahu indeks [X, Y] berada pada baris ke berapakah dari atas, baris ke berapakah dari bawah, kolom ke berapakah dari kiri, dan kolom ke berapakah dari kanan. Setelah mendapatkan keempat angka tersebut, bagaimana cara mengetahui angka pada indeks [X, Y]? Perhatikan bahwa angka pada indeks ini adalah nilai minimal

dari keempat angka tersebut.

Berikut ini merupakan contoh kode solusi dalam bahasa Pascal dan C++. Perhatikan bahwa suatu indeks [X, Y] di sebuah array 2 dimensi berukuran N x M berada pada baris ke-(X) dari atas, baris ke-(N-X+1) dari bawah, kolom ke-(Y) dari kiri, dan kolom ke-(M-Y+1) dari kanan.

```
var
  N, M, X, Y, ans : longint;
begin
  readln(N, M, X, Y);
  ans := min(X, Y);
  ans := min(ans, N-X+1);
  ans := min(ans, M-Y+1);
  writeln(ans);
end.
```

```
#include <bits/stdc++.h>
using namespace std;

int N, M, X, Y, ans;

int main() {
  scanf("%d%d%d%d", &N, &M, &X, &Y);
  ans = min(X, Y);
  ans = min(ans, N-X+1);
  ans = min(ans, M-Y+1);
  printf("%d\n", ans);
}
```

► Hubungan Kakek - Cucu

Pada soal ini, Anda diminta untuk memroses silsilah dinasti kerajaan KITOKI. Setiap lelaki dinasti ini diberi nomor 1, 2, ..., N. Pendiri dinasti kerajaan diberi nomor 1.

Anda diberi daftar hubungan ayah-anak, dengan A[Y] = X menyatakan bahwa X adalah ayah dari Y. Dari hubungan ayah-anak, Anda dapat mende-

teksi hubungan kakek-cucu karena jika Z adalah ayah dari Y, dan Y adalah ayah dari X, maka Z adalah kakek dari X.

Dari sebuah daftar hubungan ayah-anak, Anda diminta untuk membuat sebuah program yang menghitung banyaknya hubungan kakek-cucu.

Format Masukan:

Masukan terdiri dari 2 baris. Baris pertama berisi bilangan bulat N. Baris kedua berisi N-1 bilangan bulat A[2], A[3], hingga A[N] sesuai penjelasan pada deskripsi soal.

Format Keluaran:

Sebuah baris yang berisi sebuah bilangan bulat yang menandakan banyaknya hubungan kakek-cucu dari data masukan yang diberikan.

Batasan:

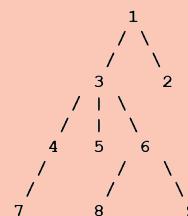
$1 \leq N \leq 10^5$
 $1 \leq A[i] \leq N$

Dijamin daftar hubungan ayah-anak membentuk silsilah yang valid

Contoh Masukan	Contoh Keluaran
9 1 1 3 3 3 4 6 6	6

Penjelasan:

Terdapat 6 hubungan kakek-cucu pada tree tersebut yakni (1-4), (1-6), (3-7), (3-8), dan (3-9).



Terdapat dua buah solusi yang akan diusulkan pada soal ini. Solusi pertama dari soal ini merupakan solusi dasar, sedangkan solusi kedua merupakan pengembangan lebih lanjut dari solusi pertama. Perhatikan bahwa materi tree dasar dan representasi graph (adjacency list) dibutuhkan dalam solusi pertama. Kedua materi tersebut merupakan beberapa materi baru yang masuk ke dalam silabus OSN mulai tahun 2017.

Untuk solusi pertama, pendekatan yang dapat dilakukan untuk menghitung banyaknya hubungan kakak-cucu adalah menghitung banyaknya cucu yang dimiliki oleh setiap orang (kemudian menjumlahkannya). Untuk menghitung banyaknya cucu dari seseorang, kita dapat menjumlahkan banyaknya anak yang dimiliki oleh anak-anak orang tersebut.

Sebagai contoh dengan kasus uji masukan

```
var
  N, ans, temp, i, j : longint;
  A : array[1..100000] of longint;
  anak : array[1..100000] of array of integer;
  anak_sz, anak_ptr : array[1..100000] of longint;

begin
  readln(N);
  for i := 2 to N do begin
    read(A[i]);
    inc(anak_sz[A[i]]);
  end;
  for i := 1 to N do begin
    setLength(anak[i], anak_sz[i]);
  end;
  for i := 1 to N do begin
    anak[A[i], anak_ptr[A[i]]] := i;
    inc(anak_ptr[A[i]]);
  end;

  for i := 1 to N do begin
    for j := Low(anak[i]) to High(anak[i]) do begin
      temp := anak[i,j];
      inc(ans, anak_sz[temp]);
    end;
  end;
  writeln(ans);
end.
```

Untuk solusi kedua, kita dapat berpikir terbalik dari solusi pertama. Alih-alih mencari tahu banyaknya cucu dari setiap orang, kita dapat menghitung berapa banyak kakak dari setiap orang. Perhatikan bahwa terdapat properti khusus dari tree yang dapat diterapkan yakni setiap orang memiliki paling banyak 1 ayah yang unik. Artinya, setiap orang memiliki paling banyak 1 kakak yang unik.

```
var
  N, A, ans, i : longint;

begin
  readln(N);
  for i := 2 to N do begin
    read(A);
    if (A <> 1) then inc(ans);
  end;
  writeln(ans);
end.
```

pada soal, banyaknya cucu dari 3 sama dengan banyaknya anak yang dimiliki 4 (yakni 1) ditambah dengan banyaknya anak yang dimiliki 5 (yakni 0) ditambah lagi dengan banyaknya anak yang dimiliki 6 (yakni 2).

Sehingga, kita dapat memanfaatkan informasi anak-anak yang dimiliki oleh seseorang. Kita akan menggunakan adjacency list untuk menyimpan informasi tersebut.

```
#include <bits/stdc++.h>
using namespace std;

int N, ans, temp;
vector<int> anak[100001];

int main()
{
  scanf("%d", &N);
  for (int i = 2; i <= N; i++) {
    scanf("%d", &temp);
    anak[temp].push_back(i);
  }
  for (int i = 1; i <= N; i++) {
    for (int j = 0; j < anak[i].size() - 1; j++) {
      temp = anak[i][j];
      ans += anak[temp].size();
    }
  }
  printf("%d\n", ans);
}
```

Sebagai contoh dengan kasus uji masukan pada soal, kakak dari 4 adalah 1 dan kakak dari 9 adalah 3. Perhatikan bahwa seseorang akan memiliki kakak jika dan hanya jika ayah dari orang tersebut bukanlah 1 karena 1 tidak memiliki ayah. Sehingga, kita hanya perlu mencari tahu banyaknya A[i] yang tidak sama dengan 1

```
#include <bits/stdc++.h>
using namespace std;

int N, A, ans;

int main()
{
  scanf("%d", &N);
  for (int i = 2; i <= N; i++) {
    scanf("%d", &A);
    if (A != 1) ans++;
  }
  printf("%d\n", ans);
}
```

Pembahasan OSN Informatika 2017

oleh Maximilianus Maria Kolbe, Alumni TOKI

► Belanja Souvenir (Day 2, Problem B)

Tag: Precompute, Sliding Window

Subsoal 1 - 3 ($2 \leq N \leq 10$)

Kita dapat menyelesaikan subsoal ini dengan melakukan brute force. Perhatikan bahwa terdapat 4 nilai yang perlu kita cari, yaitu a, b, c, dan d yang merepresentasikan rentang toko-toko yang dikunjungi oleh Kwik dan Kwak. Dengan kata lain, apabila kita mencoba semua kemungkinan nilai a, b, c, dan d yang mungkin, kita dapat memperoleh jawaban yang kita inginkan. Kemudian, kita butuh 1 kali lagi iterasi untuk memastikan apakah souvenir dalam rentang a hingga b dan c hingga d sudah memenuhi persyaratan yang diinginkan. Kompleksitas: $O(N^5)$

Subsoal 4 ($2 \leq N \leq 150$)

Perhatikan bahwa kita tidak perlu memeriksa semua kemungkinan dari a, b, c, dan d. Hal ini dikarenakan panjang rentang dari a hingga b harus sama dengan dari c hingga d. Dengan kata lain, alih-alih kita melakukan iterasi terhadap a, b, c, dan d, kita dapat melakukan iterasi terhadap M (panjang rentang), a, dan c. Kemudian kita dapat memperoleh nilai $b-a+M-1$ sekaligus dengan nilai $d=c+M-1$. Kompleksitas: $O(N^4)$

Subsoal 5 - 6 ($2 \leq N \leq 4.000$)

Karena kita hanya diminta untuk mencari satu saja konfigurasi yang maksimal, maka pada dasarnya kita tidak perlu mengiterasi semua kemungkinan yang ada. Dengan ini, kita dapat melakukan prekomputasi untuk mengetahui nilai M maksimum yang dapat kita peroleh dari SATU rentang. Sebut saja array $S[x]$ memiliki nilai M maksimum apabila awal dari rentang tersebut adalah x. Kita dapat melakukan ini dalam $O(N^2)$, dengan mengiterasi awal dan akhir dari suatu rentang.

Dengan prekomputasi tersebut, kita bisa mengetahui berapakah nilai M maksimum yang dapat diperoleh pada awal suatu rentang dalam $O(1)$. Kita dapat menggunakan data tersebut untuk mengecek rentang kedua dari yang diminta dalam $O(1)$, sehingga kita cukup mengiterasi rentang pertamanya saja dengan kompleksitas $O(N^2)$. Caranya adalah sebagai berikut : apabila rentang kiri terletak pada rentang $[a,b]$, maka $M=b-a+1$. Kemudian kita periksa apakah nilai $S[b+1]$ yang mewakili maksimum nilai M yang mungkin pada rentang kanan, lebih besar atau sama dengan $b-a+1$. Jika iya, maka nilai M tersebut menjadi calon jawaban, dan kita cukup mencari nilai yang paling besar dari calon jawaban tersebut. Kompleksitas : $O(N^2)$.

Subsoal 7 ($2 \leq N \leq 2.000.000$)

Ide nya sama persis dengan subsoal sebelumnya, tetapi kita harus mengoptimasi iterasi $O(N^2)$ yang ada pada solusi untuk subsoal sebelumnya. Terdapat beberapa hal yang dapat kita observasi :

1. Pada saat kita ingin mencari nilai M maksimum pada suatu awal rentang, apabila anggota dari rentang tersebut sudah tidak unik (suatu anggota muncul lebih dari 1 kali) maka tidak ada gunanya untuk kita melanjutkan pencarian tersebut karena selebihnya dari itu merupakan solusi yang tidak valid. Sehingga kita bisa langsung melanjutkan pencarian untuk nilai awal rentang selanjutnya.
2. Pada saat kita mulai mencari nilai M maksimum pada awal rentang yang baru, kita tidak perlu mengulangi iterasi kita dari awal, tetapi cukup melanjutkan iterasi yang ada dari awal rentang sebelumnya. Hal ini dikarenakan data dari rentang yang ada sudah pasti unik, jadi kita tidak perlu mengulangi pemeriksaan kembali.

Melihat 2 observasi di atas, dapat kita lihat bahwa kita dapat melakukan iterasi nilai awal dan akhir rentang tersebut secara bersamaan, yang menghasilkan suatu solusi dengan kompleksitas $O(N)$. Algoritma ini juga sering disebut dengan Sliding Window. Dengan cara yang sama, kita dapat menggunakan Sliding Window pada pencarian rentang pertama, kemudian memeriksa apakah rentang tersebut valid menggunakan array S yang sebelumnya sudah diprekomputasi. Kompleksitas: $O(N)$.



HALL OF FAME

TOKI 2017

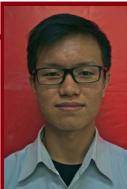
Sergio Vieri (SMA Intan Permata Hati Surabaya) telah mengikuti IOI 2016 di Rusia dan berhasil mendapatkan medali perunggu bagi Indonesia. Sergio mulai belajar pemrograman sejak kelas 6 SD mengenal Visual Basic dari ayahnya. Sergio telah meraih medali perak OSN 2014 dan medali emas OSN 2015.

Sergio merasa senang karena diberi kesempatan kedua untuk dapat mengikuti IOI 2017. Setelah belajar dari pengalaman IOI sebelumnya, Sergio pun berharap agar dapat memenangkan medali emas bagi Indonesia.



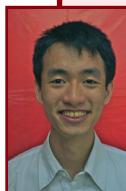
Steven Wijaya (SMA Kolese Kanisius, Jakarta) mulai belajar pemrograman PHP dari kelas 10. Setelah lulus OSN, Steven baru belajar mengenai pemrograman kompetitif dibantu kakak kelasnya di sekolah. Pada OSN 2016, Steven berhasil mendapatkan medali perunggu. Dengan konsistensinya dalam belajar, Steven berhasil menembus 4 besar pelatnas dan berangkat menuju IOI.

Pada IOI 2017, Steven akan mengalami IOI untuk pertama kalinya. Dalam menghadapi perlombaan dengan tingkat internasional ini, Steven mengaku stay calm and keep cool. Steven pun berharap dapat memberikan yang terbaik pada ajang bergengsi ini.



Kwee Lung Sin (SMA Karangturi, Semarang) merupakan teman seperjuangan Sergio Vieri. Lung Sin memulai belajar pemrograman sejak duduk di kelas 9 bersama kakak kelasnya. Tahun lalu, Lung Sin berhasil mendapatkan medali perunggu IOI 2016 yang diadakan di Rusia.

Dalam menghadapi IOI keduanya, Lung Sin merasa lebih semangat dan lebih percaya diri untuk mengerjakan permasalahan yang diberikan. Hal ini disebabkan Lung Sin telah mengetahui suasana IOI sehingga dapat mempersiapkan diri lebih awal. Dengan adanya improvisasi materi dan problem yang diberikan pada saat pelatnas, Lung Sin pun memiliki target untuk mendapatkan medali emas bagi Indonesia pada IOI 2017.



Yehezkiel Raymundo Theodoroes (SMA Katolik Rajawali, Makassar), mulai belajar bahasa pemrograman Pascal ketika duduk di kelas 9 di sekolahnya. Ketika kelas 10, Yehezkiel bertemu dengan alumni sekolohnya dalam sebuah lomba di Surabaya, sejak itu Yehezkiel mendapat bimbingan dan berkembang pesat hingga berhasil meraih medali emas pada OSN 2016. Yehezkiel juga memiliki hobi menyanyi, kegiatan pelatnas menjadi tambah meriah dengan penampilannya.

Dalam IOI 2017 ini, Yehezkiel akan berusaha keras meraih medali terbaik demi mengharumkan nama Indonesia di Kancah Internasional.

