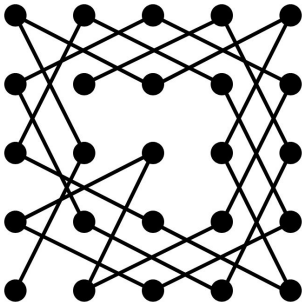
Pengenalan Pemrograman Kompetitif

COMTRAN (Competitive Traveling Nerd)

www.comtran.club

Mengenai COMTRAN



COMTRAN atau Competitive Traveling Nerd adalah komunitas pecinta pemrograman kompetitif di Universitas Telkom yang mempelajari berbagai macam algoritma dan struktur data dengan fokus kompetisi pemrograman kompetitif. Kami merasa hal ini menyenangkan dan menantang.

Kata “Competitive” menunjukkan fokus utama kami. Kata “Traveling” merujuk kepada target kami yaitu pergi ke final dan memenangkan piala. Kata “Nerd” adalah bagaimana orang-orang menyebut kami. Logo COMTRAN secara garis besar merupakan salah satu solusi dari Knight’s Tour Puzzle pada 5x5 papan catur.

Mengenai Kursus Ini

Kursus terbuka ini diharapkan dapat menjadi panduan bagi mahasiswa/i Universitas Telkom yang memiliki minat lebih terhadap algoritma dan struktur data yang juga memiliki semangat untuk mengikuti kompetisi pemrograman kompetitif.

Mengenai “Coach”

Berhubung kurangnya minat mahasiswa/i terhadap pemrograman kompetitif, maka saya, Wisnu Adi Nurcahyo (IF 2016) mengambil inisiatif untuk membuat rangkaian belajar ini. Saya tidak benar-benar menjadi “coach” karena apa yang saya buat ini demi keuntungan pribadi yaitu agar belajar lebih banyak lagi algoritma serta stuktur data sehingga dapat lebih sering lagi latihan.

Saya pernah mengikuti OSK Komputer dan berakhir di posisi tujuh. Saat SMA, saya pernah menjadi finalis pemrograman kompetitif nasional. Saat kursus ini dibuat, paling tidak saya pernah mengikuti ACM-ICPC Regional Asia Jakarta bersama-sama dengan rekan satu tim saya yaitu Ki Ageng Satria Pamungkas dan Setyo Nugroho.

Prasyarat

Sebelum mengikuti lebih jauh, pastikan kamu sudah memenuhi prasyarat di bawah karena saya mengasumsikan kamu sudah memilikinya.

1. Dapat dan merasa nyaman menulis kode bahasa pemrograman C++ dan Java.
2. Telah/sedang mengambil mata kuliah Dasar Algoritma Pemrograman.
3. Memiliki keinginan untuk lebih dekat dengan matematika.
4. Senang dengan tantangan.
5. Komitmen dalam hati.

Jargon Umum Pemrograman Kompetitif

- **OJ** atau **Online Judge** merupakan sistem *auto grading* yang menilai solusi kita secara otomatis.
- **Problem** merupakan istilah yang merujuk kepada soal atau masalah yang akan dicarikan solusinya.
- **WA** atau **Wrong Answer** merupakan kondisi dimana solusi kita tidak sesuai dengan *output*-nya.
- **AC** atau **Accepted** merupakan kondisi disaat solusi kita sesuai dengan *output*-nya.
- **TLE** atau **Time Limit Exceeded** merupakan kondisi dimana solusi kita membutuhkan waktu yang lebih lama dari batas waktu yang telah ditentukan.
- **RTE** atau **Run Time Error** merupakan kondisi dimana solusi kita mengalami masalah tak terduga saat sedang dinilai oleh OJ. Seperti contohnya **index out of range** dari suatu *array*.

Jargon Umum Pemrograman Kompetitif

- **Testcase** merupakan istilah yang merujuk pada kumpulan *input* dan *output* pengujian solusi.
- **Submit** merupakan istilah yang merujuk pada pengumpulan solusi.
- **One & Go** merupakan kondisi dimana solusi kita AC tepat saat percobaan pertama (*first try*).
- **WF** atau **World Final** adalah kompetisi pemrograman kompetitif yang tingkatannya paling tinggi.

Daftar OJ Untuk Berlatih

1. HackerRank
2. Codechef
3. SPOJ
4. TopCoder
5. PandaOJ
6. UVA
7. Codeforces
8. Open Kattis
9. TLX (Indonesia, TOKI)
10. dan masih banyak lagi...

Kenapa Berlatih Pemrograman Kompetitif

1. Dapat mempelajari banyak algoritma dan struktur yang tidak pernah diajarkan dalam bangku perkuliahan.
2. Dapat belajar untuk melakukan implementasi solusi ke dalam kode bahasa pemrograman.
3. Dapat menambah pengetahuan matematika yang berdampingan dengan ilmu komputer.
4. Dapat belajar bekerja dalam tim.
5. Dapat belajar untuk melakukan *debug* secara cepat dan tepat.
6. Dapat dengan mudah lolos dalam *technical interview*.
7. Dan yang paling penting, ini menyenangkan!

Cara Menyelesaikan Masalah Secara Umum

1. Baca deskripsi permasalahannya.
    - Selalu perhatikan format *input* dan *output*-nya.
2. Memahami abstraksi masalahnya.
3. Menyusun algoritma solusinya.
4. Implementasi dan *debug*.
    - Pastikan semua *testcase* teratasi.
5. Submit.
6. AC!
    - Jika tidak, kembali ke nomor empat.

Kesalahan Umum yang Terjadi

1. Lupa melakukan inisialisasi *variable*.
2. Penggunaan ukuran tipe data yang kurang tepat.
3. Lupa menambahkan *increment* di dalam *while*.
4. Lupa memberikan *break point* dalam *looping*.
5. Lupa mengatur *compiler* yang digunakan.
6. Melakukan pemanggilan *out of range index* pada *array*.

Gambaran Kategori Problem Secara Umum

1. Ad Hoc
2. Matematika (Aljabar Linear, Probabilitas, Teori Game, dsb)
3. String (Pattern Matching, Dictionary, dsb)
4. Teori Bilangan (Aritmatika Modulus, Teori Fermat, dsb)
5. Greedy
6. Dynamic Programming
7. Teori Graf (BFS, DFS, MST, dsb)
8. Struktur Data (Stack, Queue, Heap, dsb)
9. dan masih banyak lagi...

Sedikit Pengenalan Latihan

Problem di HackerRank (Staircase).

https://www.hackerrank.com/challenges/staircase/problem

Apa yang Harus Dilakukan?

1. Lihat **Sample Input, Sample Output**, dan **Explanation** (jika ada).
2. Lihat dan ingat **Constraint**. Jika tidak ada, biasanya masukannya tidak akan besar.
3. Apabila kurang yakin, baca **Description**.
4. Susun algoritma solusinya.
5. Submit.
6. AC!

Memahami Problem Staircase

Terdapat sebuah masukan yang berupa bilangan bulat positif tidak nol yaitu N. Dari masukan tersebut akan dikeluarkan keluaran berupa “tangga” yang dibuat dari tagar (#). Misal, N = 4.

```
   #
  ##
 ###
####
```

Perhatikan pola dari tangga tersebut. Sebelum tagar, terdapat *space*. *Space* (S) dan tagar (T) tersebut memiliki pola yaitu |S| = 3 dan |T| = 1 di baris pertama, |S| = 2 dan |T| = 2 di baris kedua, dan begitu seterusnya. Dari analisa ini, kita dapatkan kode seperti di *slide* selanjutnya.

Solusi Problem Staircase

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

int main() {
  int n;
  scanf("%d", &n);

  for (int i = 0; i < n; i++) {
      for (int j = 1; j < n - i; j++) printf(" ");
      for (int j = 0; j < i + 1; j++) printf("#");
      printf("\n");
  }

  return 0;
}
```

Sebuah Pemikiran

Kita dapat menyelesaikan Problem Staircase dengan sangat mudah, bukan? Masukan yang ada tidaklah besar sehingga kita dapat melakukan *brute-force* untuk solusinya. Lantas, apa yang terjadi apabila masukannya (dengan kata lain nilai N) mencapai 10^{12}? Hal yang pasti kita akan mendapatkan TLE. Bagaimana cara mengatasi hal tersebut? Kamu akan tahu setelah mengikuti kursus ini.

Usahakanlah untuk selalu memikirkan kemungkinan terburuknya.

Pertanyaan atau Koreksi

Jika terdapat pertanyaan atau koreksi dari *slide* ini kami persilakan kamu untuk menghubungi kami melalui surel <info@comtran.club>.

Fin