

LAPORAN PRAKTIKUM 1
Analisis algoritma



Disusun oleh :

Sina Mustopa
140810180017

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
2020

Pendahuluan

Stable Matching Problem (SMP) adalah problem algoritmik yang memberikan ilustrasi mengenai berbagai tema yang dipelajari di analisis algoritma ini. Algoritma ini muncul dari beberapa problem praktis. Oleh karena itu supaya problemnya jelas dan penyelesaian tepat perlu dilakukan 3 langkah berikut:

- Mencermati problem
- Memformulasikan problem
- Mendesain algoritma

Stable Matching Problem berasal, sebagian, pada tahun 1962, ketika David Gale dan Lloyd Shapley, dua matematika ekonom, mengajukan pertanyaan:

Bisakah seseorang merancang sebuah perguruan tinggi proses penerimaan, atau proses perekrutan pekerjaan, itu mandiri (otomatis)?

Inti dari proses aplikasi adalah interaksi antara dua jenis pihak yang berbeda: **perusahaan dan pelamar**.

Setiap pelamar memiliki daftar preferensi perusahaan yang ingin dimasuki, dan setiap perusahaan setelah aplikasi masuk-membentuk daftar preferensi akan pelamarnya. Berdasarkan preferensi ini, perusahaan memberikan penawaran kepada beberapa pelamar mereka, pelamar memilih penawaran mana yang akan mereka terima.

Bagaimana jika tidak dilakukan secara otomatis? Kemungkinan resiko kecurangan tinggi.

Jadi inilah pertanyaan yang diajukan Gale dan Shapley: Diberikan seperangkat preferensi di antara pemberi kerja dan pelamar, dapatkah kami menetapkan pelamar untuk pemberi kerja sehingga untuk setiap pemberi kerja E , dan setiap pelamar A yang tidak dijadwalkan bekerja untuk E , setidaknya satu dari dua hal berikut ini yang terjadi?

- (i) E lebih memilih setiap satu dari daftar pelamar yang diterima (A); atau
- (ii) A lebih suka situasinya saat ini daripada bekerja untuk pemberi kerja E

Jika ini berlaku, hasilnya stabil: kepentingan pribadi individu akan mencegah kesepakatan pemohon/pemberi kerja dibuat dibalik layar. Gale dan Shapley mengembangkan solusi algoritmik yang tajam untuk problem ini, yang akan kita pelajari.

Studi Kasus

SMP ini dapat dilihat juga sebagai problem menyusun sistem dimana setiap pria dan wanita akhirnya bisa berpasangan.

Jadi pertimbangkan satu set $M = \{N_1, \dots, N_n\}$ dari n pria, dan satu set $W = \{w_1, \dots, w_n\}$ dari n wanita. Produk kartesius $M \times W$ menunjukkan set dari semua pasangan bentuk yang mungkin dipesan (N, w) , di mana $N \in M$ dan $w \in W$.

Matching S adalah seperangkat pasangan yang dipesan, masing-masing dari $M \times W$, dengan properti yang masing-masing anggota M dan setiap anggota W muncul di paling banyak satu pasangan di S .

Dipandu oleh motivasi awal kita dalam hal pemberi kerja dan pelamar, kita harus khawatir tentang situasi berikut: Ada dua pasangan (N, w) dan (N', w') dalam S (seperti yang digambarkan pada Gambar 1.1) dengan properti bahwa N lebih suka w' daripada w , dan w' lebih suka N ke N' . Dalam hal ini, tidak ada yang bisa menghentikan N dan w' meninggalkan pasangan mereka saat ini dan

pergi bersama; set pernikahan menjadi tidak self-enforcing.

Tujuan kita adalah mengembalikan serangkaian pasangan tanpa ketidakstabilan (harus stabil).

Kita akan mengatakan bahwa S stabil jika

- (1) Perfect (1 laki-laki tepat berhubungan dengan satu perempuan), dan
- (2) tidak ada ketidakstabilan sehubungan dengan S

Syarat:

- Perfect Match: semua orang dicocokkan secara monogami.
 - (1) Setiap pria mendapatkan satu wanita.
 - (2) Setiap wanita mendapatkan satu pria.
- Stable Matching: pencocokan sempurna tanpa pasangan tidak stabil.

Stable Matching Problemnya:

Dengan daftar preferensi pria dan wanita, temukan sebuah stable matching jika ada.

Contoh 1

- Pertanyaan: Jika dipasangkan X-C, Y-B, dan Z-A, apakah stabil?

	favorite ↓			least favorite ↓
	1st	2nd	3rd	
Xavier	Amy	Bertha	Clare	
Yancey	Bertha	Amy	Clare	
Zeus	Amy	Bertha	Clare	

Men's Preference Profile

	favorite ↓			least favorite ↓
	1st	2nd	3rd	
Amy	Yancey	Xavier	Zeus	
Bertha	Xavier	Yancey	Zeus	
Clare	Xavier	Yancey	Zeus	

Women's Preference Profile

- Jawaban: Tidak. Bertha & Xavier akan putus

Contoh 2

- Pertanyaan: Jika dipasangkan X-A, Y-B, dan Z-C, apakah stabil?

	favorite ↓			least favorite ↓
	1st	2nd	3rd	
Xavier	Amy	Bertha	Clare	
Yancey	Bertha	Amy	Clare	
Zeus	Amy	Bertha	Clare	

Men's Preference Profile

	favorite ↓			least favorite ↓
	1st	2nd	3rd	
Amy	Yancey	Xavier	Zeus	
Bertha	Xavier	Yancey	Zeus	
Clare	Xavier	Yancey	Zeus	

Women's Preference Profile

- Jawaban: Ya

Worksheet 01

Jika Anda belum mengerjakan worksheet 01 di kelas, maka Anda dapat mengerjakannya di awal praktikum. Anda diberikan waktu 30 menit untuk menyelesaikan persoalan pada worksheet 01. Bagi Anda yang sudah mengerjakan, Anda dapat langsung mengerjakan tugas praktikum dan mencocokkan hasil worksheet 01 Anda dengan tugas praktikum.

Worksheet 01

Dengan Algoritma Gale-Shapley, cari himpunan stable-matching yang sesuai dengan preference-lists berikut ini. Gunakan processor terhebat yang Anda miliki (otak) untuk mengikuti algoritma G-S dan output tidak perlu diuraikan per-looping tetapi Anda harus memahami hasil setiap looping.

Men's Preferences Profile					
	0 th	1 st	2 nd	3 rd	4 th
Victor	Bertha	Amy	Diane	Erika	Clare
Wyatt	Diane	Bertha	Amy	Clare	Erika
Xavier	Bertha	Erika	Clare	Diane	Amy
Yancey	Amy	Diane	Clare	Bertha	Erika
Zeus	Bertha	Diane	Amy	Erika	Clare

Women's Preferences Profile					
	0 th	1 st	2 nd	3 rd	4 th
Amy	Zeus	Victor	Wyatt	Yancey	Xavier
Bertha	Xavier	Wyatt	Yancey	Victor	Zeus
Clare	Wyatt	Xavier	Yancey	Zeus	Victor
Diane	Victor	Zeus	Yancey	Xavier	Wyatt
Erika	Yancey	Wyatt	Zeus	Xavier	Victor

```
Initially all  $m \in M$  and  $w \in W$  are free
While there is a man  $m$  who is free and hasn't proposed to
every woman
  Choose such a man  $m$ 
  Let  $w$  be the highest-ranked woman in  $m$ 's preference list
  to whom  $m$  has not yet proposed
  If  $w$  is free then
    ( $m, w$ ) become engaged
  Else  $w$  is currently engaged to  $m'$ 
    If  $w$  prefers  $m'$  to  $m$  then
       $m$  remains free
    Else  $w$  prefers  $m$  to  $m'$ 
      ( $m, w$ ) become engaged
       $m'$  becomes free
    Endif
  Endif
Endwhile
Return the set  $S$  of engaged pairs
```

Tugas Praktikum

- Ubahlah pseudocode algoritma G-S pada worksheet 01 ke dalam program menggunakan bahasa C++
- Gunakan table pria sebagai table acuan untuk memudahkan Anda menentukan pasangannya.
- Cocokkan jawaban Anda pada worksheet 01 dengan hasil program yang Anda buat
- Jika ada yang berbeda tuliskan bagian mana yang berbeda dan analisislah (Poin ini disampaikan pada bagian Analisis Algoritma) yang sudah disiapkan.

Proses	Man	Woman	Single
1	Victor	Bertha	
2	Wyatt	Diane	
3	Xavier	Bertha	Victor
4	Victor	Amy	
5	Yancey	Amy	Yancey
6	Yancey	Diane	Wyatt
7	Wyatt	Bertha	Wyatt
8	Wyatt	Amy	Wyatt
9	Wyatt	Clare	
10	Zeus	Bertha	Zeus
11	Zeus	Diane	Yancey
12	Yancey	Clare	Yancey
13	Yancey	Bertha	Yancey
14	Yancey	Erika	

Analisis Algoritma

Jawablah pertanyaan berikut:

1. Apakah jawaban Anda di Worksheet 01 dan Program sama persis? Jika Tidak? Kenapa?

Iya, jawaban dari worksheet 01 dan hasil program adalah sama.

Anda diminta untuk membuktikan algoritma G-S benar dengan menjawab pertanyaan berikut:

Fakta (1.1):

Seorang wanita tetap bertunangan dari titik di mana dia menerima proposal pertamanya; dan urutan mitra yang bertunangan dengannya menjadi lebih baik dan lebih baik lagi (hal ini sesuai dengan daftar preferensi wanita). → tidak perlu dipertanyakan

Fakta (1.2):

Urutan wanita yang dilamar pria lebih buruk dan lebih buruk lagi (hal ini sesuai dengan daftar preferensi pria). → tidak perlu dipertanyakan

Teorema (1.3):

Algoritma G-S berakhir setelah paling banyak n^2 iterasi menggunakan While Loop. Buktikan!

Ketika menggunakan proses while loop syntax memiliki kemajuan, pria lajang melamar wanita pilihan selanjutnya dalam daftar pilihannya, lalu seseorang yang belum pernah ia lamar sebelumnya. Karena ada n jumlah laki-laki dan setiap daftar preferensi memiliki panjang sebesar n , ada paling banyak n^2 ajuan lamaran sebesar n^2 yang dapat terjadi. Jadi jumlah dalam iterasi yang bisa terjadi paling banyak adalah n^2 .

Teorema (1.4):

Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita yang belum dia ajak bertunangan.

Buktikan!

Pembuktian ini bisa dilakukan dengan bukti berdasarkan kontradiksi. Misalkan ada waktu pada saat tertentu dalam pelaksanaan algoritma ketika seorang pria lajang (single), namun telah mengusulkan kepada setiap wanita yang ada. Ini artinya, setiap wanita telah diusulkan setidaknya satu kali usulan. Dengan teori 1, mendapatkan bahwa setiap wanita bertunangan. Jadi, kita telah melibatkan n wanita dan karenanya n laki-laki bertunangan, yang tertulis bahwa m juga terlibat bertentangan dengan asumsi bahwa m adalah lajang (single).

Teorema (1.5):

Himpunan S yang dikembalikan saat terminasi adalah *perfect matching*

Buktikan!

Karena untuk setiap laki-laki haruslah berpasangan dengan setiap wanita yang ada.

Teorema (1.6):

Sebuah eksekusi algoritma G-S mengembalikan satu set pasangan S . Set S adalah pasangan yang stabil.

Buktikan!

Menunjukkan bahwa pencocokan yang dikembalikan adalah pencocokan sempurna, pencocokan untuk masing-masing mendapatkan pasangan. Buktinya dengan kontradiksi. Misalkan tidak, maka ada seorang pria yang masih lajang (single) di akhir algoritma. Menurut teori ke-2, itu berarti m belum melamar beberapa wanita yang ada. Selanjutnya, algoritma tidak akan keluar dari pengulangan loop, menghasilkan kontradiksi yang diinginkan.

Menunjukkan bahwa pencocokan yang dikembalikan stabil. Buktinya dengan kontradiksi. Misalkan ada laki-laki m dan m' dan wanita w dan w' sehingga (m, w) dan (m', w') berada di S , tetapi m lebih suka w' ke w , dan w' lebih suka m ke m' . Dengan algoritma, w adalah wanita terakhir yang diajukan. Karena m lebih suka w' ke w , m harus sudah bisa mengusulkan ke w' sebelum usulannya ke wanita w . Pada saat itu, w' bertunangan dengan seorang pria, katakanlah m'' , yang ia sukai lebih dari pria m . Pada akhirnya, w' bertunangan dengan m' . Oleh teori 1, menemukan bahwa w' lebih memilih m' daripada m'' dan lebih memilih m'' daripada m ; ini dituliskan bahwa w' lebih suka m' daripada m , bertentangan dengan asumsi bahwa w' lebih memilih m daripada m' .

```

1  /*
2  Nama : Sina Mustopa
3  NPM : 140810180017
4  Kelas : A
5  Deskripsi : Stable Matching Problem pada Kasus pasangan
6  */
7
8  #include <string.h>
9  #include <stdio.h>
10 #include <iostream>
11 #define N 5
12
13 using namespace std;
14
15
16 bool chooseMen(int prioritas[2 * N][N], int w, int m, int m1)
17 {
18     for (int i = 0; i < N; i++)
19     {
20         if (prioritas[w][i] == m1)
21             return true;
22
23         if (prioritas[w][i] == m)
24             return false;
25     }
26 }
27 void stable(int prioritas[2 * N][N])
28 {
29     int pasanganWanita[N];
30
31     bool priaJomblo[N];
32
33     memset(pasanganWanita, -1, sizeof(pasanganWanita));
34     memset(priaJomblo, false, sizeof(priaJomblo));
35     int jumlahJomblo = N;

```

```

37     while (jumlahJomblo > 0)
38     {
39         int m;
40         for (m = 0; m < N; m++)
41             if (priaJomblo[m] == false)
42                 break;
43
44         for (int i = 0; i < N && priaJomblo[m] == false; i++)
45         {
46             int w = prioritas[m][i];
47
48             if (pasanganWanita[w - N] == -1)
49             {
50                 pasanganWanita[w - N] = m;
51                 priaJomblo[m] = true;
52                 jumlahJomblo--;
53             }
54
55             else
56             {
57                 int m1 = pasanganWanita[w - N];
58
59                 if (chooseMen(prioritas, w, m, m1) == false)
60                 {
61                     pasanganWanita[w - N] = m;
62                     priaJomblo[m] = true;
63                     priaJomblo[m1] = false;
64                 }
65             }
66         }
67     }
68     cout << "===== " << endl;
69     cout << "  Pria ----- Wanita  " << endl;
70     cout << "===== " << endl;
71     string man;

```

```

72     string woman;
73     for (int i = 0; i < N; i++)
74     {
75         if (i < N)
76         {
77             if (pasanganWanita[i] == 0)
78                 man = "victor";
79             if (pasanganWanita[i] == 1)
80                 man = "wyatt";
81             if (pasanganWanita[i] == 2)
82                 man = "xavier";
83             if (pasanganWanita[i] == 3)
84                 man = "yancey";
85             if (pasanganWanita[i] == 4)
86                 man = "zeus";
87
88             if (i == 0)
89                 woman = "amy";
90             if (i == 1)
91                 woman = "bertha";
92             if (i == 2)
93                 woman = "clare";
94             if (i == 3)
95                 woman = "diane";
96             if (i == 4)
97                 woman = "erika";
98         }
99         cout << " " << man << "\t\t" << woman << endl;
100     }
101     cout << "-----" << endl;
102 }
103

```

```

104 int main()
105 {
106     int prioritas[2 * N][N] = {{6, 5, 8, 9, 7},
107                                {8, 6, 5, 7, 9},
108                                {6, 9, 7, 8, 5},
109                                {5, 8, 7, 6, 9},
110                                {6, 8, 5, 9, 7},
111                                {4, 0, 1, 3, 2},
112                                {2, 1, 3, 0, 4},
113                                {1, 2, 3, 4, 0},
114                                {0, 4, 3, 2, 1},
115                                {3, 1, 4, 2, 0}};
116     stable(prioritas);
117
118     return 0;
119 }
120

```

Teknik Pengumpulan

- Lakukan push ke github/gitlab untuk semua program dan laporan hasil analisa yang berisi jawaban dari pertanyaan-pertanyaan yang diajukan. Silahkan sepakati dengan asisten praktikum.

Penutup

- Ingat, berdasarkan Peraturan Rektor No 46 Tahun 2016 tentang Penyelenggaraan Pendidikan, mahasiswa wajib mengikuti praktikum 100%
- Apabila tidak hadir pada salah satu kegiatan praktikum segeralah minta tugas pengganti ke asisten praktikum
- Kurangnya kehadiran Anda di praktikum, memungkinkan nilai praktikum Anda tidak akan dimasukkan ke nilai mata kuliah.