TUGAS 1

ANALISIS ALGORITMA

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

2019

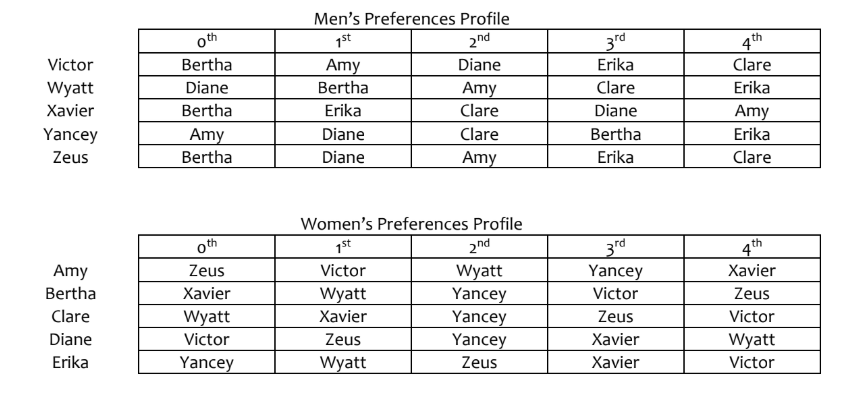
DISUSUN OLEH:

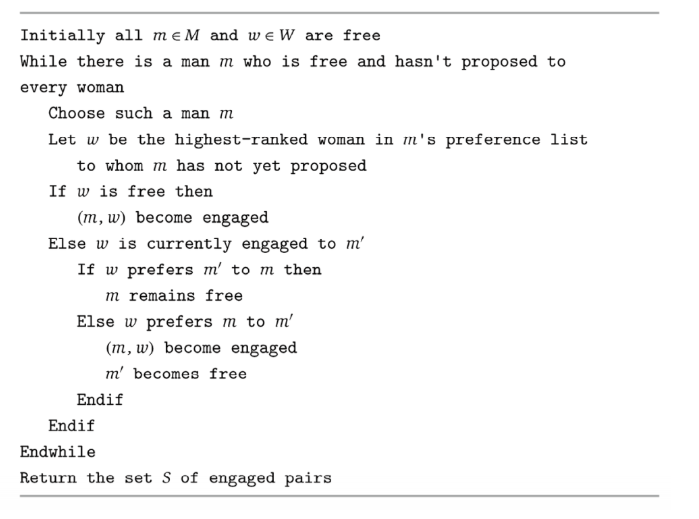
AITHRA JUNIA BOUTY

140810180054

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.





### 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 analisalah (Poin ini disampaikan pada bagian Analisis Algoritma) yang sudah disiapkan.

**Menggunakan c++**

/\*

Nama : Aithra Junia Bouty

Kelas : B

NPM : 140810180054

\*/

#include <iostream>

#include <string.h>

#include <stdio.h>

using namespace std;

#define N 5

bool chooseMen(int prioritas[2 \* N][N], int w, int m, int m1)

{

for (int i = 0; i < N; i++)

{

if (prioritas[w][i] == m1)

return true;

if (prioritas[w][i] == m)

return false;

}

}

void smp(int prioritas[2 \* N][N]) //smp = stable matching problem

{

int women[N];

bool men[N];

memset(women, -1, sizeof(women));

memset(men, false, sizeof(men));

int jumlahJomblo = N;

while (jumlahJomblo > 0)

{

int m;

for (m = 0; m < N; m++)

if (men[m] == false)

break;

for (int i = 0; i < N && men[m] == false; i++)

{

int w = prioritas[m][i];

if (women[w - N] == -1)

{

women[w - N] = m;

men[m] = true;

jumlahJomblo--;

}

else

{

int m1 = women[w - N];

if (chooseMen(prioritas, w, m, m1) == false)

{

women[w - N] = m;

men[m] = true;

men[m1] = false;

}

}

}

}

cout << "-----------------" << endl;

cout << " Man Women " << endl;

cout << "-----------------" << endl;

string man;

string woman;

for (int i = 0; i < N; i++)

{

if (i < N)

{

if (women[i] == 0)

woman = "Amy";

if (women[i] == 1)

woman = "Bertha";

if (women[i] == 2)

woman = "Clare";

if (women[i] == 3)

woman = "Diane";

if (women[i] == 4)

woman = "Erika";

if (i == 0)

man = "Victor";

if (i == 1)

man = "Wyatt";

if (i == 2)

man = "Xavier";

if (i == 3)

man = "Yancey";

if (i == 4)

man = "Zeus";

}

cout << " " << man << "\t " << woman << endl;

}

cout << "-----------------" << endl;

}

int main()

{

int prioritas[2 \* N][N] = {{6, 5, 8, 9, 7},

{8, 6, 5, 7, 9},

{6, 9, 7, 8, 5},

{5, 8, 7, 6, 9},

{6, 8, 5, 9, 7},

{4, 0, 1, 3, 2},

{2, 1, 3, 0, 4},

{1, 2, 3, 4, 0},

{0, 4, 3, 2, 1},

{3, 1, 4, 2, 0}};

smp(prioritas);

return 0;

}

Result:

**Iterasi 1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| 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 |

**Iterasi 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| 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 |

**Iterasi 3**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| 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 |

**Iterasi 4**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| 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 |

**Iterasi 5**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 0th | 1st | 2nd | 3rd | 4th |
| 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 |

### Analisis Algoritma

Jawablah pertanyaan berikut:

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

Ya, sama persis.

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 n2 iterasi menggunakan While Loop. Buktikan!

Karena pada saat while loop, pria yang tidak memiliki pasangan wanita melamar wanita pada pilihan pertamanya, apabila tidak dimiliki pria lain dan dipilih oleh si wanita lebih dari pasangannya saat itu. Karena pria berjumlah n, sama dengan panjang preferensi wanita, maka lamaran paling banyak ada n2. Jadi, jumlah iterasi paling banyak adalah n2.

#### Teorema (1.4):

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

Buktikan!

Karena apabila semua wanita sudah dilamar setidaknyaoleh satu pria saja, maka, n wanita yang dilamar menyebabkan n pria sudah bertunangan. Jadi, apabila ada pria yang masih lajang pada waktu tertentu algoritma dijalankan, berarti ada seorang wanita yang belum diajak bertunangan.

#### Teorema (1.5):

Himpunan S yang dikembalikan saat terminasi adalah *perfect matching* Buktikan!

Karena semua pria mendapatkan pasangan wanitanya.

#### 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. Buktinya dengan kontradiksi. Misalkan tidak, maka ada seorang pria yang masih lajang di akhir algoritma. Menurut teori 2, itu berarti m belum melamar beberapa wanita. Tetapi kemudian, algoritma tidak akan keluar dari pengulangan loop, menghasilkan kontradiksi yang diinginkan.

Menunjukkan bahwa pencocokan yang dikembalikan stabil. Lagi-lagi 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 saya ajukan. Karena m lebih suka w’ ke w, m harus sudah mengusulkan ke w’ sebelum usulannya ke w. Pada saat itu, atau nanti, w’ bertunangan dengan seorang pria, katakanlah m’’, yang ia sukai lebih dari 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 menyiratkan bahwa w’ lebih suka m’ daripada m, bertentangan dengan asumsi bahwa w’ lebih memilih m daripada m’.