LAPORAN PRAKTIKUM 1

ANALISIS ALGORITMA



DISUSUN OLEH:

NAMA : RAISSA AMINI

NPM : 140810180073

Program Studi S-1 Teknik Informatika

Departemen Ilmu Komputer

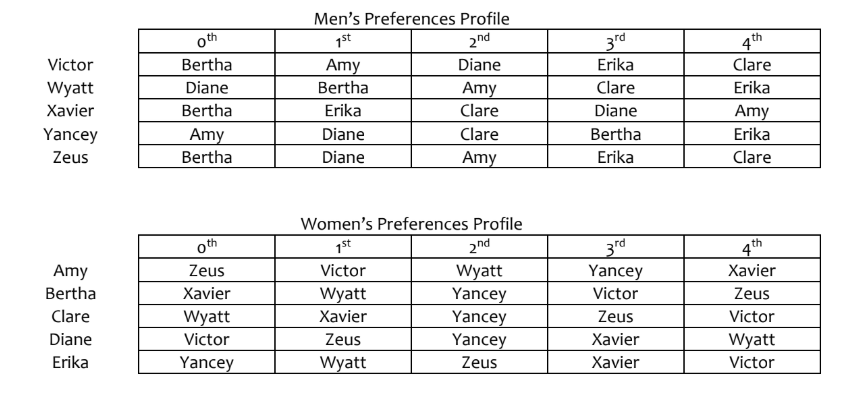
Fakultas Matematika dan Ilmu Pengetahuan Alam

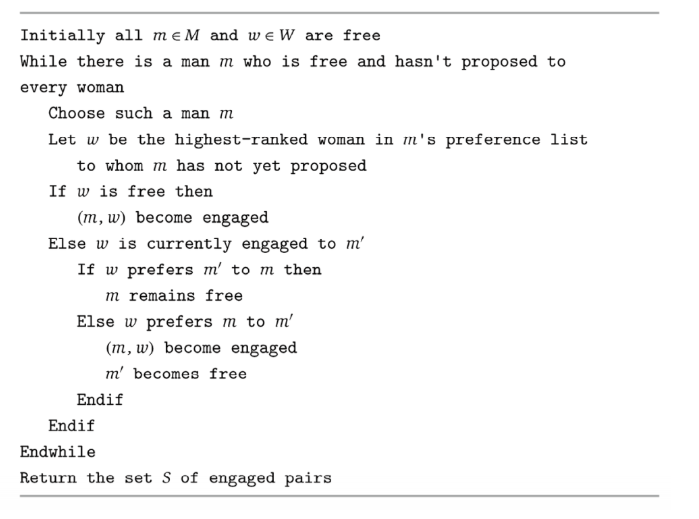
Universitas Padjadjaran

2018

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++**

#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 adalah untuk pencocokan nya

{

int pasanganWanita[N];

bool priaJomblo[N];

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

memset(priaJomblo, false, sizeof(priaJomblo));

int jumlahJomblo = N;

while (jumlahJomblo > 0)

{

int m;

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

if (priaJomblo[m] == false)

break;

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

{

int w = prioritas[m][i];

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

{

pasanganWanita[w - N] = m;

priaJomblo[m] = true;

jumlahJomblo--;

}

else

{

int m1 = pasanganWanita[w - N];

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

{

pasanganWanita[w - N] = m;

priaJomblo[m] = true;

priaJomblo[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 (pasanganWanita[i] == 0)

man = "Victor";

if (pasanganWanita[i] == 1)

man = "Wyatt";

if (pasanganWanita[i] == 2)

man = "Xavier";

if (pasanganWanita[i] == 3)

man = "Yancey";

if (pasanganWanita[i] == 4)

man = "Zeus";

if (i == 0)

woman = "Amy";

if (i == 1)

woman = "Bertha";

if (i == 2)

woman = "Clare";

if (i == 3)

woman = "Diane";

if (i == 4)

woman = "Erika";

}

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},

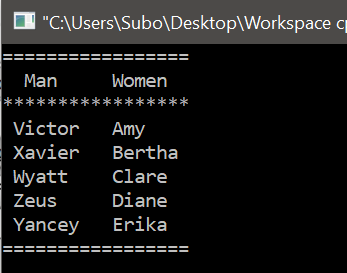
{0, 4, 3, 2, 1},

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

smp(prioritas);

return 0;

}



**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**

Victor-Amy

Wyatt-Clare

Xavier-Bertha

Yancey-Erika

Zeus-Diane

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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

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!

Pada saat while loop pasangan dicoba untuk dicocokan, yaitu dimana pria yang lajang melamar wanita berikutnya dalam daftar pilihannya, lalu seseorang yang belum pernah dilamar oleh pria tersebut sebelumnya. Karena laki-laki berjumlah n dan setiap daftar prefensi panjangnya adalah n, maka paling banyak proposal adalah n2. Jadi jumlah dalam iterasi paling banyak adalah n2

Algoritma

#### Teorema (1.4):

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

Buktikan!

Misalkan ada waktu tertentu dalam pelaksanaan algoritma ketika seorang pria lajang, namun telah melamar setiap wanita. Ini berarti, setiap wanita telah diusulkan setidaknya satu kali, dan setiap wanita bertunangan. Jadi, kita telah melibatkan n wanita dan karenanya n laki-laki bertunangan, yang menyiratkan bahwa m juga terlibat dan tidak mungkin ada pria yang tidak memiliki pasangan karena jumlah pasangannya sama-sama berjumlah n.

#### 

#### Teorema (1.5):

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

Karena semua mendapatkan pasangan

#### 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’.