**Tugas Praktikum**

**Analisis Algortima**

****

Disusun oleh :

Muhammad Fakhri Rahman (140810170026)

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

JATINANGOR

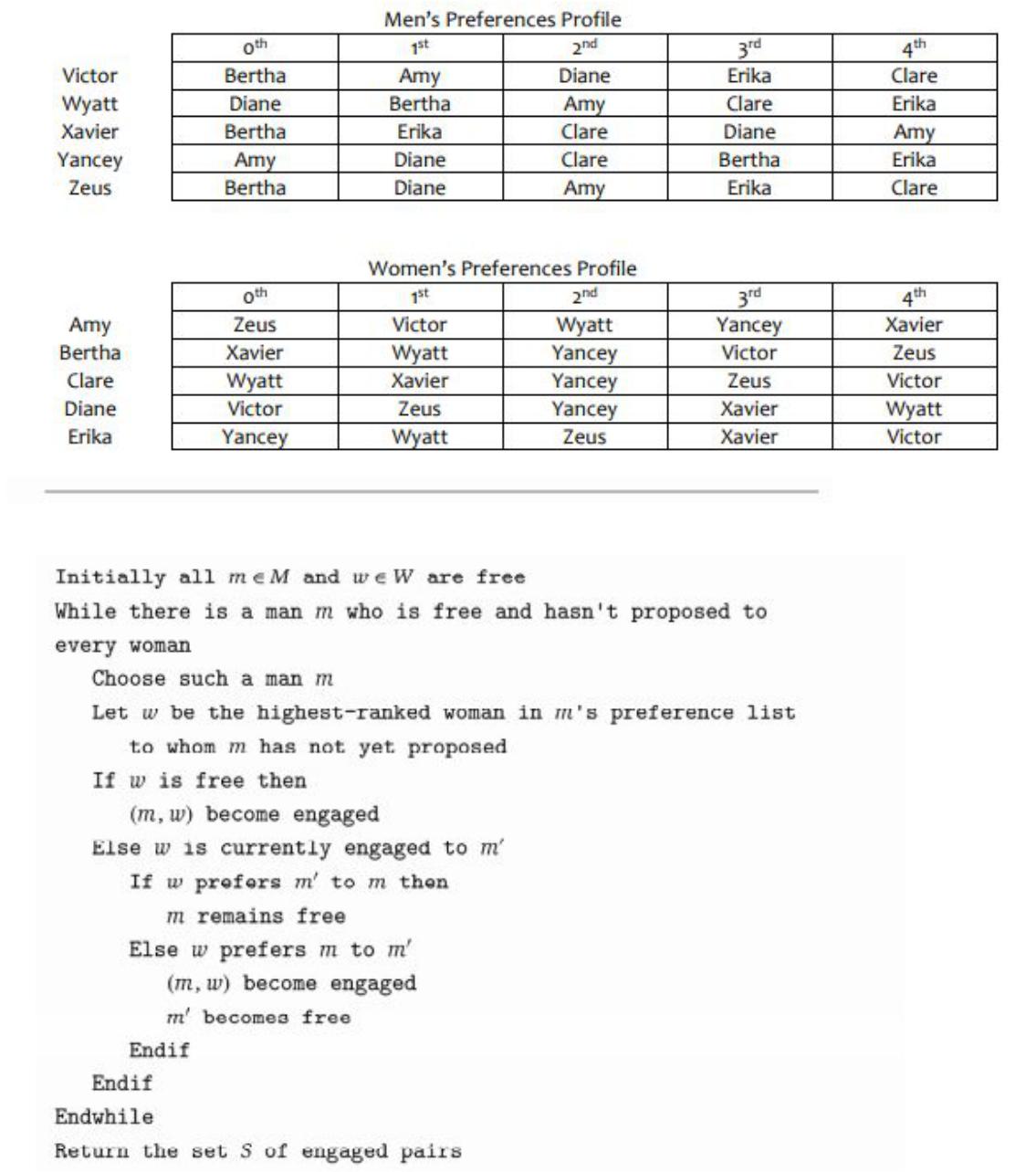
2019

Worksheet 01

Jika Anda belum mengerajakan 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.



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.

Analisis Algoritma

* tracing manual dan codingan terlampir Jawablah pertanyaan berikut:

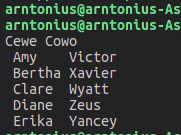
1. Apakah jawaban Anda di Worksheet 01 dan Program sama persis? Jika Tidak? Kenapa?

Sama hasilnya ataupun output nya menghasilkan pasangan yang benar antara Worksheet 1 dengan program

hasil :

tracing manual Program

(Victor, Amy)

(Wyatt, Clare)

(Xavier, Bertha)

(Yancey, Erika)

(Zeus, Diane)

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

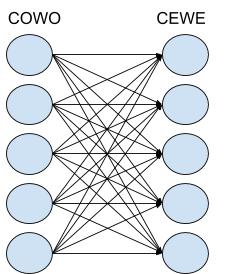
**Teorema (1.3):**

2​

Algoritma G-S berakhir setelah paling banyak n​ iterasi menggunakan While Loop.

Buktikan!

Benar , karena setiap pasangan akan dicocokkan dengan setiap lawan pasangannya dan ketika belum ada kecocokan akan terus berulang sehingga paling banyak akan menghasilkan paling banyak 16 kali loop (dengan 4 pasangan).



akan diperoleh looping maksimal yaitu 4^2 dengan beberapa ketentuan yang berbeda. Jumlah perulangan akan berkurang apabila tingkat prioritas satu dengan yang lain tidak saling berbentrokan.

**Teorema (1.4):**

Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita

yang belum dia ajak bertunangan.

Buktikan!

Betul karena setiap laki-laki akan dipasangkan dengan pasangan yang ditentukan dengan prioritas tertinggi dari pria. Ketika beberapa orang yang ada belum mendapat pasangan maka berarti pria tersebut gagal berhubungan sehingga dia menjadi single dan ketika masih single berarti ada wanita yang masih single pula yang belum dia temui. contohnya :

Jika ada 2 pria bebas dan 3 pria yang telah bertunangan dari total keseluruhan 5 pria, maka pasti ada 2 wanita yang belum diajak bertunangan yang tidak berada di prioritas utama dari

kedua pria tersebut. Maka, apabila terdapat seorang pria yang tidak memiliki pasangan pasti ada juga wanita yang tidak memiliki pasangan, karena pada algoritma Gale-Shapley mengharuskan setiap pria berpasangan dengan wanita.

**Teorema (1.5):**

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

Benar, karena dalam algoritma ini setiap pria mendapatkan pasangan seorang wanita, sehingga dapat dikatakan perfect matching. Meskipun pemasangan ini dilakukan dengan adanya beberapa kali perulangan yang dikarenakan urutan prioritas yang tidak sama antara pria dan wanita, tapi tetap dikatakan perfect matching karena tiap pria mendapatkan pasangan wanita, begitupun sebaliknya.

**Teorema (1.6):**

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

yang stabil.

Buktikan!

Yaa , karena sesuai dengan kecocokan nya setiap pasangan akan mengalami tingkat pencocokan yang berbeda dengan beberapa pasangan yang ada. Pada algoritma G-S ini tentunya akan mencocokan dengan beberapa kali loop namun akan selalu menghasilkan sebuah data yang stabil dari awal pencocokan hingga di akhir. contohnya :

Asumsikan himpunan S memiliki ketidakstabilan. Secara khusus, ada pria dan wanita m, w, m0, dan W0 seperti itu bahwa semua hal berikut ini benar: (m, w) **∈** / S

(m, w0) **∈** S

(m0, w) **∈** S

Pm (w) <Pm (w0)

Pw (m) <Pw (m0).

Dengan asumsi ini, m dan w akan bertunangan dan memutuskan keterlibatan mereka dengan w0 dan m0. Untuk bukti ini, akan mempertimbangkan apa yang harus terjadi untuk menyebabkan ketidakstabilan ini dan menunjukkan bahwa semua kasus mengarah pada kontradiksi. Pertama-tama kita akan mengamati bahwa kita harus melamar m, dan pada akhirnya akan bertunangan. Untuk m0 yang dia sukai kurang dari m prioritasnya. Satu-satunya cara dia bisa melamar m0, dia harus melamar setiap pria yang lebih disukainya daripada pria yang mencakup m.

**Tracing Manual**

Step 1   
m = victor

Victor->bertha

Bertha free

(victor,bertha)

Step 2

m = Wyatt

wyatt -> Diane

Diane free

(wyatt,diane)

Step 3

m = Xavier

Xavier-> bertha

Bertha engaged to m’=victor

Bertha prefers Xavier than victor

(Xavier,bertha) // victor free

Step 4

m = yancey

yancey -> amy

amy free

(yancey, amy)

Step 5

m = zeus

zeus -> bertha

bertha engaged to m’=Xavier

bertha prefers Xavier than zeus

zeus remains free

step 6

m = victor

victor -> amy

amy engaged to m’ = yancey

amy prefers victor than yancey

(victor, amy)

Yancey becomes free

Step 7

m = yancey

yancey -> diane

diane engaged to m’ = wyatt

diane prefers yancey than wyatt

(yancey, diane)

Wyatt becomes free

Step 8

m = zeus

zeus -> diane

diane engaged to m’ = yancey

diane prefers zeus than yancey

(zeus, diane)

Yancey becomes free

Step 9

m = wyatt

wyatt -> amy

amy engaged to m’ = victor

amy prefers victor than wyatt

wyatt remains free

step 10

m = yancey

yancey -> clare

clare free

(yancey, clare)

Step 11

m = wyatt

wyatt -> clare

clare engaged to m’ = yancey

clare prefers wyatt than yancey

(wyatt, clare)

Step 12

m = yancey

yancey -> Erika

Erika free

(yancey, Erika)

Stable matching :

**(victor, amy), (wyatt, clare), (Xavier, bertha), (yancey, Erika), (zeus, diane)**

**Codingan**

#include <iostream>

#include <string.h>

#include <stdio.h>

using namespace std;

#define N 5

bool pPilihL1dariL(int pilih[2\*N][N], int p, int l, int l1)

{

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

{

if (pilih[p][i] == l1)

return true;

if (pilih[p][i] == l)

return false;

}

}

void stabil(int pilih[2\*N][N])

{

int pPasangan[N];

bool lKosong[N];

memset(pPasangan, -1, sizeof(pPasangan)); memset(lKosong, false, sizeof(lKosong)); int freecount = N;

while (freecount > 0)

{

int l;

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

if (lKosong[l] == false)

break;

for (int i = 0; i < N && lKosong[l] == false; i++)

{

int p = pilih[l][i];

if (pPasangan[p-N] == -1)

{

pPasangan[p-N] = l;

lKosong[l] = true;

freecount--;

}

else

{

int l1 = pPasangan[p-N];

if (pPilihL1dariL(pilih, p, l, l1) == false)

{

pPasangan[p-N] = l;

lKosong[l] = true;

lKosong[l1] = false;

}

}

}

}

cout << "Cewe Cowo" << endl;

string namco;

string namce;

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

{

if(i<N){

if(pPasangan[i]==0)

namco="Victor";

if(pPasangan[i]==1)

namco="Wyatt";

if(pPasangan[i]==2)

namco="Xavier";

if(pPasangan[i]==3)

namco="Yancey";

if(pPasangan[i]==4)

namco="Zeus";

if(i==0)

namce="Amy";

if(i==1)

namce="Bertha";

if(i==2)

namce="Clare";

if(i==3)

namce="Diane";

if(i==4)

namce="Erika";

}

cout << " " << namce << "\t" << namco << endl;

}

}

int main()

{

int pilih[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}

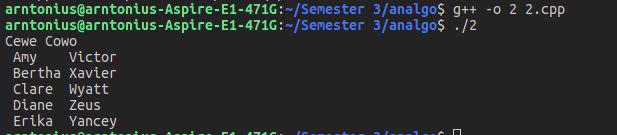
};

stabil(pilih);

return 0;

}

**Output**

****