**TUGAS PRAKTIKUM ANALISIS ALGORITMA**

**Modul Praktikum 1**



**Muhammad Afif**

**140810170045**

**Kelas: A**

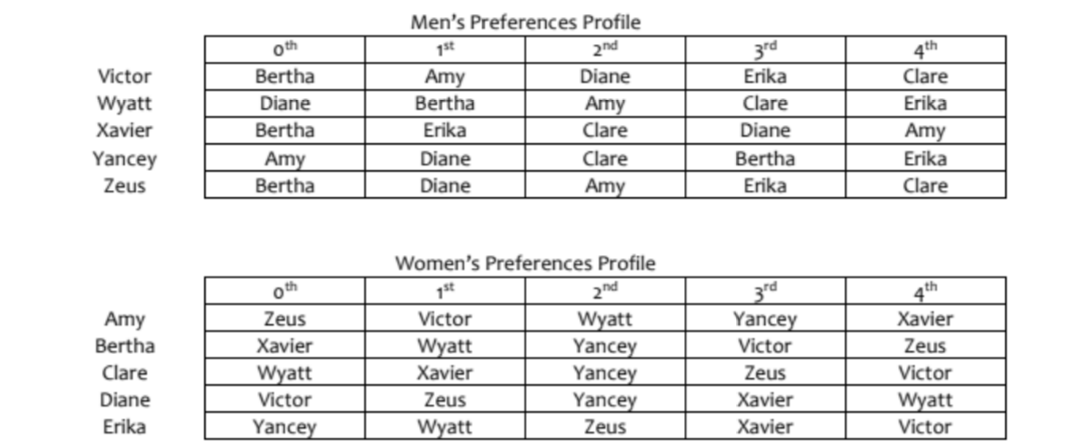
**JURUSAN S-1 TEKNIK INFORMATIKA**

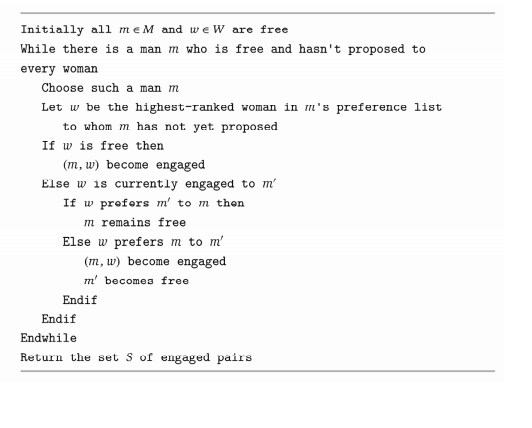
**FAKULTAS MATEMATIKA dan ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**Worksheet 1**

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.





Jawaban

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Victor | Bertha | Amy |  |  |  |
| Wyatt | Dianne | Bertha | Amy | Clare |  |
| Xavier | Bertha |  |  |  |  |
| Yancy | Amy | Dianne | Clare | Bertha | Erika |
| Zeus | Bertha | Dianne |  |  |  |

**Victor -> Amy**

**Wyatt -> Clare**

**Xavier -> Bertha**

**Yancey -> Erika**

**Zeus -> Diane**

**Tugas**

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.

C++:

// C++ program for stable marriage problem

#include <iostream>

#include <stdio.h>

using namespace std;

// Number of Men or Women

#define N 5

// This function returns true if woman 'w' prefers man 'm1' over man 'm'

bool wPrefersM1OverM(int prefer[2\*N][N], int w, int m, int m1)

{

// Check if w prefers m over her current engagment m1

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

{

// If m1 comes before m in lisr of w, then w prefers her

// cirrent engagement, don't do anything

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

return true;

// If m cmes before m1 in w's list, then free her current

// engagement and engage her with m

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

return false;

}

}

void assignName(int i, int wPartner){

cout << " " ;

if(i==0){

cout<<"Amy";

}else if(i==1){

cout<<"Bertha";

}else if(i==2){

cout<<"Clare";

}else if(i==3){

cout<<"Diane";

}else if(i==4){

cout<<"Erika";

}else

cout<<" ";

cout<<"\t\t";

if(wPartner==0){

cout<<"Victor";

}else if(wPartner==1){

cout<<"Wyatt";

}else if(wPartner==2){

cout<<"Xavier";

}else if(wPartner==3){

cout<<"Yancey";

}else if(wPartner==4){

cout<<"Zeus";

}else

cout<<" ";

cout<<endl;

}

// Prints stable matching for N boys and N girls. Boys are numbered as 0 to

// N-1. Girls are numbereed as N to 2N-1.

void stableMarriage(int prefer[2\*N][N])

{

// Stores partner of women. This is our output array that

// stores paing information. The value of wPartner[i]

// indicates the partner assigned to woman N+i. Note that

// the woman numbers between N and 2\*N-1. The value -1

// indicates that (N+i)'th woman is free

int wPartner[N];

// An array to store availability of men. If mFree[i] is

// false, then man 'i' is free, otherwise engaged.

bool mFree[N];

// Initialize all men and women as free

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

memset(mFree, false, sizeof(mFree));

int freeCount = N;

// While there are free men

while (freeCount > 0)

{

// Pick the first free man (we could pick any)

int m;

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

if (mFree[m] == false)

break;

// One by one go to all women according to m's preferences.

// Here m is the picked free man

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

{

int w = prefer[m][i];

// The woman of preference is free, w and m become

// partners (Note that the partnership maybe changed

// later). So we can say they are engaged not married

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

{

wPartner[w-N] = m;

mFree[m] = true;

freeCount--;

}

else // If w is not free

{

// Find current engagement of w

int m1 = wPartner[w-N];

// If w prefers m over her current engagement m1,

// then break the engagement between w and m1 and

// engage m with w.

if (wPrefersM1OverM(prefer, w, m, m1) == false)

{

wPartner[w-N] = m;

mFree[m] = true;

mFree[m1] = false;

}

} // End of Else

} // End of the for loop that goes to all women in m's list

} // End of main while loop

// Print the solution

cout << "Woman Man" << endl;

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

cout << " ";

assignName(i,-1);

assignName(-1,wPartner[i]);

cout<<endl;

}

}

// Driver program to test above functions

int main()

{

int prefer[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},

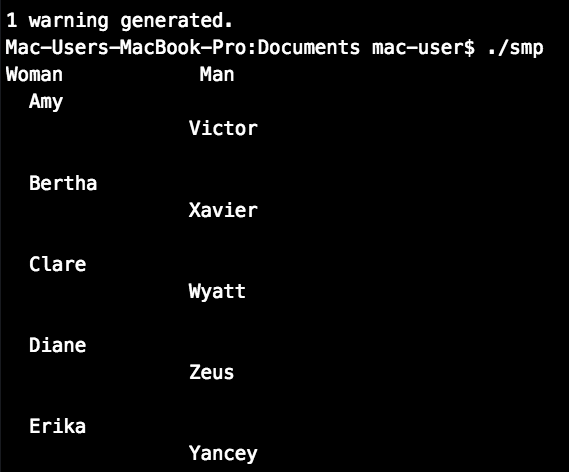
};

stableMarriage(prefer);

return 0;

}

Screenshot

****

**Analisis Algoritma**

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



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!



Teorema (1.4):

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

yang belum dia ajak bertunangan.

Buktikan!



Teorema (1.5):

Himpunan S yang dikembalikan saat terminasi adalah perfect matching

Buktikan!



Teorema (1.6):

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

Buktikan!

