**Laporan Praktikum Analisis Algoritma**

**Modul Praktikum 1**



**Disusun Oleh:**

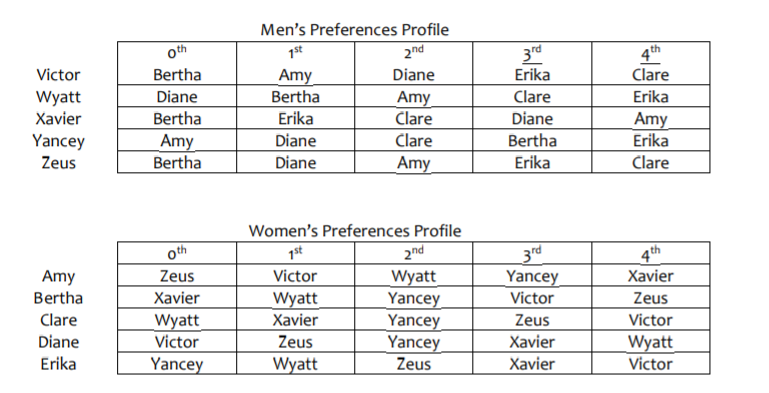
Muhammad Fahmi Alwan (140810170052)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS PADJADJARAN**

**2018/2019**

1. Worksheet 1

**Menggunakan Algoritma Gale-Shapley cari himpunan stable-matching yang sesuai preferensi list diatas.**

1. Seluruh pria masih masih dan wanita masih belum mempunyai pasangan
2. Choose Victor

If Bertha is free then (Victor, Bertha)

Engaged =

(Victor, Bertha)

1. Choose Wyatt

If Diane is free then (Wyatt, Diane)

Engaged =

(Victor, Bertha)

(Wyatt, Diane)

1. Choose Xavier

If Bertha is free (FALSE)

Else Bertha currently engaged to Victor

If Bertha prefers Victor (FALSE)

Else Bertha prefers Xavier then (Xavier, Bertha) and Victor free

Engaged =

(Wyatt, Diane)

(Xavier, Bertha)

1. Choose Yancey

If Amy is free then (Yancey, Amy)

Engaged =

(Wyatt, Diane)

(Xavier, Bertha)

(Yancey, Amy)

1. Choose Zeus

If Bertha is free (FALSE)

Else Bertha is currently engaged to Xavier

If Bertha prefers Xavier to Zeus (TRUE) then Zeus free

Engaged =

(Wyatt, Diane)

(Xavier, Bertha)

(Yancey, Amy)

1. Choose Victor

If Amy is free (FALSE)

Else Amy is currently engaged to Yancey

If Amy prefers Yancey to Victor (FALSE)

Else Amy prefers Victor to Yancey then (Victor, Amy) and Yancey free

Engaged =

(Wyatt, Diane)

(Xavier, Bertha)

(Victor, Amy)

1. Choose Wyatt

If Bertha is free (FALSE)

Else Bertha is currently engaged to Xavier

If Bertha prefers Xavier to Wyatt (TRUE) then Wyatt is free

Engaged =

(Wyatt, Diane)

(Xavier, Bertha)

(Victor, Amy)

1. Choose Yancey

If Diane is free (FALSE)

Else Diane is currently engaged to Wyatt

If Diane prefers Wyatt to Yancey (FALSE)

Else Diane prefers Yancey to Wyatt then (Yancey, Diane) and Wyatt is free

Engaged =

(Xavier, Bertha)

(Victor, Amy)

(Yancey, Diane)

1. Choose Zeus

If Diane is free (FALSE)

Else Diane is currently engaged to Yancey

If Diane prefers Yancey to Zeus (FALSE)

Else Diane prefers Zeus to Yancey then (Zeus, Diane) and Yancey is free

Engaged =

(Xavier, Bertha)

(Victor, Amy)

(Zeus, Diane)

1. Choose Wyatt

If Amy is free (FALSE)  
Else Amy is currently engaged to Victor

If Amy is prefers Victor (TRUE) then Wyatt is free

Engaged =

(Xavier, Bertha)

(Victor, Amy)

(Zeus, Diane)

1. Choose Yancey

If Clare is free then (Yancey, Clare)

Engaged =

(Xavier, Bertha)

(Victor, Amy)

(Zeus, Diane)

(Yancey, Clare)

1. Choose Wyatt

If Clare is free (FALSE)

Else Clare is currently engaged to Yancey

If Clare is prefers to Yancey (FALSE)

Else Clare is prefers Wyatt to Yancey then (Wyatt, Clare)

Engaged =

(Xavier, Bertha)

(Victor, Amy)

(Zeus, Diane)

(Wyatt, Clare)

1. Choose Yancey

If Erika is free then (Yancey, Erika)

Engaged =

(Xavier, Bertha)

(Victor, Amy)

(Zeus, Diane)

(Wyatt, Clare)

(Yancey, Erika)

1. Source Code

/\* NAMA : MUHAMMAD FAHMI ALWAN

NPM : 140810170052

KELAS : B

JUDUL : GALE SHAPLEY

DIBUAT : 5/3/19

\*/

#include <algorithm>

#include <iostream>

#include <map>

#include <queue>

#include <string>

#include <vector>

using namespace std;

const char \*men\_data[][6] = {

{ "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" }

};

const char \*women\_data[][6] = {

{ "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" }

};

typedef vector<string> PrefList;

typedef map<string, PrefList> PrefMap;

typedef map<string, string> Couples;

// Does 'first' appear before 'second' in preference list? \*membedakan list pertama sebagai yg punya preferensi dengan preferensinya

bool prefers(const PrefList &prefer, const string &first, const string &second)

{

for (PrefList::const\_iterator it = prefer.begin(); it != prefer.end(); ++it)

{

if (\*it == first) return true; // orang yg punya preferensi

if (\*it == second) return false; // preferensinya

}

return false; // tidak ada preferensi

int main()

{

PrefMap men\_pref, women\_pref;

queue<string> bachelors;

// init data structures

for (int i = 0; i < 5; ++i) // orangnya

{

for (int j = 1; j < 6; ++j) // preferensinya

{

men\_pref[ men\_data[i][0]].push\_back( men\_data[i][j]);

women\_pref[women\_data[i][0]].push\_back(women\_data[i][j]);

}

bachelors.push(men\_data[i][0]);

}

Couples engaged; // <perempuan, pria>

cout << "Proses Matchmaking:\n"; //tampil proses untuk mendapatkan pasangan

while (!bachelors.empty())

{

const string &suitor = bachelors.front();

const PrefList &preflist = men\_pref[suitor];

for (PrefList::const\_iterator it = preflist.begin(); it != preflist.end(); ++it)

{

const string &bride = \*it;

if (engaged.find(bride) == engaged.end()) // perempuannya masih kosong

{

cout << "\t" << bride << " dan " << suitor << "\n";

engaged[bride] = suitor; // berpasangan

break;

}

const string &groom = engaged[bride];

if (prefers(women\_pref[bride], suitor, groom))

{

cout << "\t" << bride << " memutus " << groom << " untuk " << suitor << "\n";

bachelors.push(groom); // ngeputusin prianya

engaged[bride] = suitor; // milih pria yang di preferensinya

break;

}

}

bachelors.pop(); // pop diakhir untuk tidak membatalkan preferensi pelamar

}

cout << "Hasil Pasangan\n"; //tampil hasil pasangan

for (Couples::const\_iterator it = engaged.begin(); it != engaged.end(); ++it)

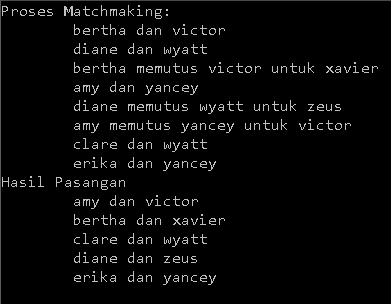
{

cout << "\t" << it->first << " dan " << it->second << "\n";

}

}

**Screenshot**



1. Analisis
2. Apakah jawaban anda di worksheet 01 dan program sama persis? Jika tidak? Kenapa?

Jawab :

Ya, sama persis.

1. Algoritma G-S berakhir setelah paling banyak n2 iterasi menggunakan While Loop. Buktikan!

Jawab :

Ya. Pada worksheet 1 kasus apabila ada satu pria yang tidak putus putus dengan pasangannya maka akan terjadi pencocokan sebanyak n kali. Maka iterasi terbanyak nya aka nana n2 kali.

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

Jawab :

Karena setiap pria dan wanita harus monogami atau memiliki satu pasangan manka jika ada satu pria yang di bebaskan maka akan ada wanita yang sama bebasnya sebanyak jumlah pria yang bebas .

1. Himpunan S yang dikembalikan saat terminasi adalah perfect matching. Buktikan!

Jawab :

Ya. Pada worksheet satu dapat dilihat hasilnya:

(Xavier, Bertha)

(Victor, Amy)

(Zeus, Diane)

(Wyatt, Clare)

(Yancey, Erika)

Setiap pria dan wanita monogami atau memiliki satu pasangan dan tidak ada pria atau wanita yang tidak memiliki pasangan. Sehingga terbukti himpunan S yang dikembalikan adalah perfect matching.

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

Jawab :

Syarat stable matching:

1. Harus perfect match

2. Setiap pasangan set S tidak ada ketidak stabilan

Pada jawaban no. 4 diketahui bahwa syarat satu detal terpenuhi. Algoritma Gale Shapley mencocokan pasangan sesuai dengan keinginannya, sehingga setiap pasangan pada set S stabil. Maka syarat kedau terpenuhi. Dengan terpenuhinya kedua syarat maka algoritma Gale Shapley merupakan stable match.