

TUGAS 1
PRAKTIKUM ANALISIS ALGORITMA



Disusun oleh :

Hasna Karimah

140810160020

PROGRAM STUDI S-1 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
SUMEDANG
2019

Algoritma Gale-Shapley

Men's Preferences Profile					
	0 th	1 st	2 nd	3 rd	4 th
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

Women's Preferences Profile					
	0 th	1 st	2 nd	3 rd	4 th
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

```

Initially all  $m \in M$  and  $w \in W$  are free
While there is a man  $m$  who is free and hasn't proposed to
every woman
    Choose such a man  $m$ 
    Let  $w$  be the highest-ranked woman in  $m$ 's preference list
    to whom  $m$  has not yet proposed
    If  $w$  is free then
         $(m, w)$  become engaged
    Else  $w$  is currently engaged to  $m'$ 
        If  $w$  prefers  $m'$  to  $m$  then
             $m$  remains free
        Else  $w$  prefers  $m$  to  $m'$ 
             $(m, w)$  become engaged
             $m'$  becomes free
        Endif
    Endif
Endwhile
Return the set  $S$  of engaged pairs

```

Code :

```

/* Hasna Karimah

Tugas Praktikum Analisis Algoritma - Worksheet 1
Stable Marriage Problem
5-3-2019
=====*/

```

```

#include <iostream>
using namespace std;

int findIndex(string name){
    int resultIndex;
    if (name == "Victor" || name == "Amy")
        resultIndex = 0;
    else if (name == "Wyatt" || name == "Bertha")
        resultIndex = 1;
    else if (name == "Xavier" || name == "Clare")
        resultIndex = 2;
    else if (name == "Yancey" || name == "Diane")
        resultIndex = 3;
    else
        resultIndex = 4;

    return resultIndex;
}

int main()
{
    string men[5][2] = {
        {"Victor", "free"},
        {"Wyatt", "free"},
        {"Xavier", "free"},
        {"Yancey", "free"},
        {"Zeus", "free"},
    };

    string menPreference[5][5] = {
        {"Bertha", "Amy", "Diane", "Erika", "Clare" }, // Victor
        {"Diane", "Bertha", "Amy", "Clare", "Erika" }, // Wyatt
        {"Bertha", "Erika", "Clare", "Diane", "Amy" }, // Xavier
        {"Amy", "Diane", "Clare", "Bertha", "Erika" }, // Yancey
        {"Bertha", "Diane", "Amy", "Erika", "Clare" } // Zeus
    };

    string women[5][2] = {
        {"Amy", "free"},
        {"Bertha", "free"},
        {"Clare", "free"},
        {"Diane", "free"},
        {"Erika", "free"},
    };

    string womenPreference[5][5] = {
        {"Zeus", "Victor", "Wyatt", "Yancey", "Xavier" }, // Amy
        {"Xavier", "Wyatt", "Yancey", "Victor", "Zeus" }, // Bertha
    };
}

```

```

        {"Wyatt", "Xavier", "Yancey", "Zeus", "Victor" }, // Clare
        {"Victor", "Zeus", "Yancey", "Xavier", "Wyatt" }, // Diane
        {"Yancey", "Wyatt", "Zeus", "Xavier", "Victor" } // Erika
    };

    string engaged[5][2] = {
        { "Amy", "?"},
        { "Bertha", "?"},
        { "Clare", "?"},
        { "Diane", "?"},
        { "Erika", "?"}
    };

    bool isSomeoneFree = true;
    int tries = 0;

    for (int i=0; i<5; i++){

        for (int j=0; j<5; j++){

            if (men[i][1] == "free"){

                if (women[ findIndex(menPreference[i][j]) ][1] == "free" ){

                    men[i][1] = "engaged";
                    women[ findIndex(menPreference[i][j]) ][1] = "engaged";
                    engaged[ findIndex(menPreference[i][j]) ][1] = men[i][0];
                    cout<<engaged[ findIndex(menPreference[i][j]) ][1] <<" with "
                        <<engaged[ findIndex(menPreference[i][j]) ][0]
                        <<" engaged!" <<endl;
                    isSomeoneFree = false;

                    break;
                }

                else {

                    for (int k=0; k<5; k++){

                        if ( womenPreference[ findIndex(menPreference[i][j]) ][k]
==
                            engaged[ findIndex(menPreference[i][j]) ][1] )
                            break;

```

```

        if ( womenPreference[ findIndex(menPreference[i][j]) ][k]
== men[i][0]
        && k < 5 ){

        cout<<engaged[ findIndex(menPreference[i][j]) ][1]
<<" with "
        <<engaged[ findIndex(menPreference[i][j]) ][0]
        <<" break apart! then ";

        men[ findIndex(engaged[
findIndex(menPreference[i][j]) ][1]) ][1] = "free";

        men[i][1] = "engaged";
        women[ findIndex(menPreference[i][j]) ][1] =
"engaged";
        engaged[ findIndex(menPreference[i][j]) ][1] =
men[i][0];

        cout<<engaged[ findIndex(menPreference[i][j]) ][1]
<<" with "
        <<engaged[ findIndex(menPreference[i][j]) ][0]
        <<" engaged!" <<endl;
        isSomeoneFree = true;
    }
    }
}
else
    break;
}

Tries++;

if (isSomeoneFree == true && i == 5 - 1 )
    i = -1;
}

if (isSomeoneFree == false){
    cout<<endl <<endl
    <<"Stable couple are : "<<endl<<endl;
    for(int i=0; i<5; i++){
        cout<<engaged[i][1] <<" engaged with "
        <<engaged[i][0] <<endl;
    }
}

```

```
    }  
}  
  
return 0;  
}
```

Output :

```
Victor with Bertha engaged!  
Wyatt with Diane engaged!  
Victor with Bertha break apart! then Xavier with Bertha engaged!  
Yancey with Amy engaged!  
Wyatt with Diane break apart! then Zeus with Diane engaged!  
Yancey with Amy break apart! then Victor with Amy engaged!  
Wyatt with Clare engaged!  
Yancey with Erika engaged!  
  
Stable couple are :  
  
Victor engaged with Amy  
Xavier engaged with Bertha  
Wyatt engaged with Clare  
Zeus engaged with Diane  
Yancey engaged with Erika
```

1. m = victor
victor → Bertha
if (Bertha == free) //True
 (Victor, Bertha)
Man free:
Wyatt
Xavier
Yancey
Zeus
Women free:
Amy
Clare
Diane
Erika
2. m = wyatt
wyatt → Diane
if (Diane == free) //true
 (wyatt, diane)
Man free:
Xavier
Yancey
Zeus
Women free:
Amy
Clare
Erika
3. m = Xavier
Xavier → Bertha
If (bertha == free) //false
Else
 If (bertha prefer victor) //false
 Else (bertha prefer Xavier)
//true
 (Xavier, bertha)
 Victor free
Man free:
Victor
Yancey
Zeus
Women free:
Amy
Clare
Erika
4. m = Yancey
Yancey → Amy
If (Amy == free) //true
 (Yancey, Amy)
Man free:
Victor
Zeus
Women free:
Clare
Erika
5. m = Zeus
Zeus → Bertha
If (Bertha == free) //false
Else
 If (bertha prefer Xavier) // true
 (Xavier, Bertha)
 Zeus free
Man free:
Victor
Zeus
Women free:
Clare
Erika
6. m = Victor
Victor → Amy
If (Amy == free) //false
Else
 If (amy prefer yancey) //false
 Else (amy prefer victor) //true
 (Victor, Amy)
 Yancey free
Man free:
Yancey
Zeus
Women free:
Clare
Erika
7. m = Zeus
Zeus → Diane
If (Diane == free) //false
Else
 If (Diane prefer wyatt) //false
 Else (Diane prefer zeus) //true

(Zeus, Diane)
Wyatt free

Man free:
Wyatt
Yancey
Women free:
Clare
Erika

8. m = Yancey
yancey → Diane
if (diane == free) //false
else
 if (diane prefer zeus) //true
 (Zeus, Diane)
 Yancey free

Man free:
Wyatt
Yancey
Women free:
Clare
Erika

9. m = wyatt
wyatt → bertha
if (bertha == free) // false
else
 if (bertha prefer Xavier) //true
 (Xavier, Bertha)
 Wyatt free

Man free:
Wyatt
Yancey
Women free:
Clare
Erika

10. m = yancey
yancey → clare
if (clare == free) //true
 (Yancey, Clare)

Man free:
Wyatt

Women free:

Erika

11. m = wyatt
wyatt → Amy
if (amy == free) //false
else
 if (amy prefer victor) //true
 (Victor, Amy)
 Wyatt free

Man free:
Wyatt

Women free:
Erika

12. m = wyatt
wyatt → Clare
if (clare == free) //false
else
 if (clare prefer yancey) //false
 else (clare prefer wyatt) //true
 (Wyatt, Clare)
 Yancey free

Man free:
Yancey

Women free:
Erika

13. m = yancey
yancey → Erika
if (Erika == free) //true
 (Yancey, Erika)

Man free:
-
Women free:
-

Jadi pasangannya :

- Victor, Amy
- Xavier, bertha
- Wyatt, Clare
- Zeus, Diane
- Yancey, Erika

Pembuktian Algoritma G-S

Anda diminta untuk membuktikan algoritma G-S benar dengan menjawab pertanyaan berikut:

1. **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

Benar. Wanita langsung bertunangan dari titik dimana dia menerima proposal pertamanya.

2. **Fakta (1.2):** Urutan wanita yang dilamar pria lebih buruk dan lebih buruk lagi (hal ini sesuai dengan daftar preferensi pria). -> tidak perlu dipertanyakan

Benar. Daftar preferensi pilihan tiap pria sudah berurut secara *descending*.

3. **Teorema (1.3):** Algoritma G-S berakhir setelah paling banyak n^2 iterasi menggunakan While Loop. Buktikan!

Paling banyak n^2 iterasi didapat dari hasil perkalian antara jumlah pria dan wanita yang ada didalam himpunan. Aturan yang menyebutkan setiap pria akan mengajukan proposal ke, maksimal, seluruh wanita yang ada memberikan pencerahan bahwa jumlah maksimal proposal yang diajukan adalah jumlah pria dikali jumlah wanita.

4. **Teorema (1.4):** Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita yang belum dia ajak bertunangan. Buktikan!

Dapat dibuktikan dengan kontradiksi. Misal ada waktu tertentu dalam pelaksanaan algoritma ketika seorang pria single, namun telah mengusulkan kepada setiap wanita. Ini berarti saat ini, setiap wanita telah dipasangkan setidaknya satu kali. Ada n wanita yang bertunangan dan ada n pria yang bertunangan, yang berarti bahwa tidak mungkin ada orang yang masih single jika setiap n sudah berpasangan.

5. **Teorema (1.5):** Himpunan S yang dikembalikan saat terminasi adalah perfect matching Buktikan!

Pria pasti hanya akan melamar apabila belum berpasangan atau pasangan sebelumnya tidak cocok. Sedangkan wanita akan selalu memilih pria dengan preferensi teratas untuk bertunangan dengannya. Baik pria dan wanita, prinsip mereka adalah melamar sesuai urutan preferensi. Dengan itu Himpunan S adalah perfect matching dikarenakan teori tersebut.

6. **Teorema (1.6):** Sebuah eksekusi algoritma G-S mengembalikan satu set pasangan S . Set S adalah pasangan yang stabil. Buktikan!

Tidak ada pria yang bisa ditolak oleh semua wanita. Wanita pun harus mendapatkan satu pria dan ia hanya dapat menolak lamaran ketika ada pria yang lebih tinggi preferensinya dibandingkan pria sebelumnya. Setiap iterasi dari loop sementara melibatkan tepat satu proposal dan pria tidak akan melamar wanita yang sama dua kali.