

Naufal Ariful Amri

140810180009

Kelas A

## Worksheet 1 :

### Stable Matching Algorithm

Dengan Algoritma Gale-Shapley, cari himpunan stable-matching yang sesuai dengan preferencelists berikut ini. Gunakan processor terhebat yang Anda miliki (otak) untuk mengikuti algoritma GS dan uraikan outputnya untuk setiap loop hingga menghasilkan stable-matching.

Men's Preferences Profile					
	0 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
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 <sup>th</sup>	1 <sup>st</sup>	2 <sup>nd</sup>	3 <sup>rd</sup>	4 <sup>th</sup>
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

Jawab :

Iterasi 1 :

	0	1	2	3	4
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
	0	1	2	3	4

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

Iterasi 2 :

	0	1	2	3	4
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

	0	1	2	3	4
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

Iterasi 3 :

	0	1	2	3	4
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

	0	1	2	3	4
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

Iterasi 4 :

	0	1	2	3	4
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

	0	1	2	3	4
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

Iterasi 5 :

	0	1	2	3	4
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

	0	1	2	3	4
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

Iterasi 6 :

	0	1	2	3	4
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

	0	1	2	3	4
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

Iterasi 7 :

	0	1	2	3	4
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

	0	1	2	3	4
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

Kesimpulan

Stable Matching Algorithm

(Victor , Amy) , (Wyatt , Clare) , (Xavier , Bertha) , (Yancey , Erika) , (Zeus , Diane)

Kodingan :

```
#include <iostream>
#include <cstring>
#include <cstdio>
#define N 5
using namespace std;
```

```

bool wPrefersM1OverM(int prefer[2*N][N], int w, int m, int m1)
{
    for (int i = 0; i < N; i++)
    {
        if (prefer[w][i] == m1)
            return true;
        if (prefer[w][i] == m)
            return false;
    }
}

void stableMarriage(int prefer[2*N][N])
{
    int wPartner[N];
    bool mFree[N];
    memset(wPartner, -1, sizeof(wPartner));
    memset(mFree, false, sizeof(mFree));
    int freeCount = N;
    while (freeCount > 0)
    {
        int m;
        for (m = 0; m < N; m++)
            if (mFree[m] == false)
                break;
        for (int i = 0; i < N && mFree[m] == false; i++)
        {
            int w = prefer[m][i];
            if (wPartner[w - N] == -1)
            {
                wPartner[w - N] = m;
                mFree[m] = true;
                freeCount--;
            }
            else
            {
                int m1 = wPartner[w - N];
                if (wPrefersM1OverM(prefer, w, m, m1) == false)
                {
                    wPartner[w - N] = m;
                    mFree[m] = true;
                    mFree[m1] = false;
                }
            }
        }
    }
    cout << "Woman   Man" << endl;
    for (int i = 0; i < N; i++)
        cout << " " << i + N << "\t" << wPartner[i] << endl;
}

/*
 * Main
 */
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;
}

```

**Output :**

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Naufal Ariful Amri> cd "d:\YOU MUST DO IT\Devcom"

Woman    Man
-----
5         0
6         2
7         1
8         4
9         3

PS D:\YOU MUST DO IT\Devcom>

```

**Analisis:**

1. Apakah jawaban Anda di Worksheet 01 dan Program sama persis? Jika Tidak? Kenapa?  
Ya, jawabannya sama persis (Beda angka dan huruf saja).

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

- Fakta (1.1): Setiap wanita (Diwakilkan dengan angka) tetap bertunangan dari titik di mana dia menerima proposal pertamanya; dan urutan prioritas 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 (Diwakilkan dengan angka)).  
→ Tidak perlu dipertanyakan
- Teorema (1.3): Algoritma G-S berakhir setelah paling banyak  $n^2$  iterasi menggunakan While Loop. Buktikan!  
→ Ketika kondisi terburuk, while loop akan mencapai  $n^2$  karena dalam looping laki-laki terdapat kemungkinan setiap laki-laki mendapatkan prioritas terakhir dan pada setiap wanita, laki-laki mendapatkan prioritas pertama.
- Teorema (1.4): Jika seorang pria bebas di beberapa titik dalam eksekusi algoritma, maka ada seorang wanita yang belum dia ajak bertunangan. Buktikan!  
→ Karena didalam algoritma, hanya dapat berpasangan dengan 1 wanita (Korespondensi 1-1). Jika laki-laki belum berpasangan, tentu saja ada relasi yang belum berkait (wanita).
- Teorema (1.5): Himpunan S yang dikembalikan saat terminasi adalah perfect matching. Buktikan!  
→ Setiap pria pasti akan melamar wanita. Sedangkan wanita akan membandingkan prioritas jika ada 2 laki-laki yang melamar. Dalam kasus ini, ketika laki-laki melamar wanita yang sudah punya pasangan sementara, wanita



bisa membandingkan prioritas. Dan dalam situasi ini merupakan perfect matching karena wanita akan mendapatkan laki-laki prioritas terbaiknya.

- 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. Dalam kasus terburuk, seorang pria bisa melamar semua wanita hingga ia mendapatkan pasangannya. Dengan menggunakan algoritma G-S, dapat kita simpulkan bahwa algoritma akan berakhir setelah semua mendapat pasangan. Selain itu pasangan yang sudah dipasangkan juga harus menunjukkan kecocokan dengan preferensi masing-masing dan pencocokan algoritma G-S dianggap selalu stabil.