

LAPORAN PRAKTIKUM 1

**MATA KULIAH
ANALISIS ALGORITMA
D10G.4205 & D10K.0400601**



**PENGAJAR : (1) MIRA SURYANI, S.Pd., M.Kom
(2) INO SURYANA, Drs., M.Kom
(3) R. SUDRAJAT, Drs., M.Si**

FAKULTAS : MIPA

SEMESTER : IV dan VI

DISUSUN OLEH : AHMAD FAAIZ A (140810180023)

**PROGRAM STUDI S-1 TEKNIK INFORMATIKA
DEPARTEMEN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
FEBRUARI 2019**

Hasil Worksheet 1

Step:

1. Victor -> Bertha
2. Wyatt -> Dianne
3. Xavier -> Bertha, Victor free
4. Yancey -> Amy
5. Zeus -> Bertha, Zeus remains free
6. Victor -> Amy, Yancey free
7. Yancey -> Diane, Wyatt free
8. Zeus -> Diane, Yancey free
9. Wyatt -> Bertha, Wyatt remains free
10. Yancey -> Clare
11. Wyatt -> Amy, Wyatt remains free
12. Wyatt -> Clare, Yancey free
13. Yancey -> Bertha, Yancey remains free
14. Yancey -> Erika

Output:

	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

	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

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

Free: -

Sudah menghasilkan Stable-Matching

Program stable-matching

```
#include <iostream>

using namespace std;

int main()
{
    int n = 5; //number of available men/women

    string men[n] = {"victor", "wyatt", "xavier", "yancey", "zeus"};
    string women[n] = {"amy", "bertha", "clare", "diane", "erika"};

    int mPref[n][n] = {
        {1, 0, 3, 4, 2},
        {3, 1, 0, 2, 4},
        {1, 4, 2, 3, 1},
        {0, 3, 2, 1, 4},
        {1, 3, 1, 4, 2},
    };
    int wPref[n][n] = {
        {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},
    };

    int mPartner[n] = {-1, -1, -1, -1, -1}; //-1 means free
    int wPartner[n] = {-1, -1, -1, -1, -1};

    int freeCount = n;

    //while there is a man who is free and hasnt proposed to every woman
    while (freeCount >= 0)
    {
        for (int m = 0; m < n; m++)
        {
            if (mPartner[m] != -1) //if m has engaged to w, pass
                continue;

            bool stopScanning = false;

            //choose such a man (m)
            //let w be the highest-
            ranked woman in m's preference list to whom m has not yet proposed
            for (int w = 0; w < n; w++) // loop through mPref
            {
                if (mPref[m][w] == -1) //if w has rejected m, pass
                    continue;

                //if w is free
                if (wPartner[mPref[m][w]] == -1)
                {
                    mPartner[m] = mPref[m][w];
                    wPartner[mPref[m][w]] = m;
                    break;
                }
            }
        }
    }
}
```

```

        //else w is currently engaged to m'
        else
        {
            int currentMan = m; // (m)
            int currentPartner = wPartner[mPref[m][w]]; // (m')

            //find index of (m) and (m') in the wPref list to show the who
            is more preferred
            for (int i = 0; i < n; i++)
            {
                //if w prefers m' to m (current partner)
                if (wPref[mPref[m][w]][i] == currentPartner) // (m') came u
                p first
                {
                    //m remains free
                    mPref[m][w] = -1; //blacklist w
                    stopScanning = true;
                    break;
                }
                //if w prefers m to m'
                if (wPref[mPref[m][w]][i] == currentMan)
                {
                    // (m,w) become engaged
                    mPartner[m] = mPref[m][w];
                    wPartner[mPref[m][w]] = m;
                    //m' becomes free
                    mPartner[currentPartner] = -1;
                    mPref[currentPartner][w] = -1; //blacklist w
                    stopScanning = true;
                    break;
                }
            }
        }
        if (stopScanning)
            break;
    }
    freeCount--;
}

//print solution
cout << "Man\tWoman" << endl;
for (int i = 0; i < n; i++)
    cout << men[i] << "\t" << women[mPartner[i]] << endl;
}

```

Output:

```

d:\Documents\UNPAD\Perkuliahan\Tugas & Praktikum\Semester 4\Analgo\Prak\Pertemuan-1>stable-matching-problem
Man      Woman
victor   amy
wyatt    clare
xavier   bertha
yancey   erika
zeus     diane

```

Analisis Algoritma

Jawablah pertanyaan berikut:

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

Ya, jawaban di worksheet 01 sama persis dengan output program.

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!

Jumlah laki – laki dan perempuan masing – masing adalah n orang yang masing masing akan berpasangan sehingga menghasilkan n pasangan (laki – laki dan perempuan) di akhir program. Setiap laki – laki memiliki preferensi sebanyak n perempuan untuk dilamar, sehingga kemungkinan terburuk jumlah iterasi yang harus dijalankan adalah $n \times n = n^2$ iterasi.

Teorema (1.4):

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

Anggaplah seorang pria bebas dan telah melamar setiap wanita. Ketika seorang wanita telah dilamar, dia tidak lagi bebas. Pilihannya adalah antara dia tetap dengan pasangannya saat ini, atau bertunangan dengan pria yang melamarnya jika pria itu lebih disukai dibanding pria pasangannya saat ini. Oleh karena itu dapat dikatakan bahwa semua wanita akan terlibat dalam proses pertunangan. Karena satu wanita dipasangkan hanya dengan satu pria, maka n pria akan bertunangan dengan n wanita. Sehingga tidak benar bahwa ada seorang wanita yang belum dia ajak bertunangan (tidak terbukti).

Teorema (1.5):

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

Seorang wanita bisa bebas, atau bertunangan dengan hanya satu pria, jadi setiap wanita hanya muncul sekali di set.

Seorang pria yang sudah bertunangan tidak akan melamar wanita lain. Seorang pria bisa bebas atau bertunangan dengan satu wanita, sehingga setiap pria hanya akan muncul sekali di set.

Oleh karena itu, set yang dikembalikan saat terminasi adalah *perfect matching* karena satu pria akan berpasangan dengan tepat satu wanita.

Teorema (1.6):

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

Dari Teorema sebelumnya, dapat dikatakan bahwa set S adalah *perfect matching*. Tidak ada pria yang bisa ditolak oleh semua wanita dan tiap wanita 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.