TUGAS 1 PRAKTIKUM

ANALISIS ALGORITMA



Muhammad Luthfiansyah

140810160023

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2020

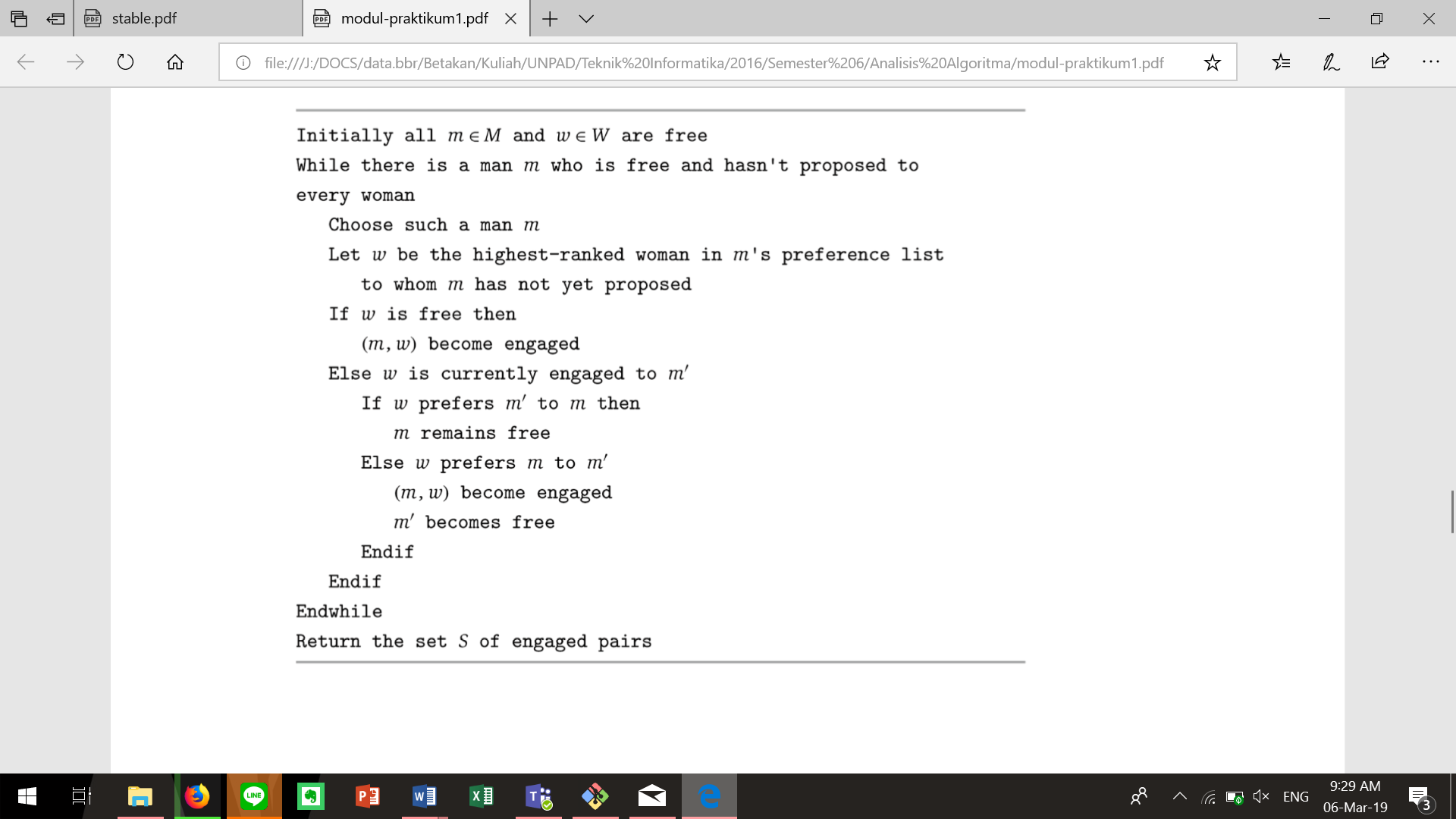
**Soal**



Hasil yang diharapkan:



Dengan pseudocodenya sebagai berikut:



Dari tabel dapat disimpulkan proses yang terjadi adalah sebagai berikut:

* Victor dengan Bertha.
* Wyatt dengan Diane.
* Xavier dengan Bertha, karena Bertha lebih menyukai Xavier sehingga Victor dilepas.
* Yancey dengan Amy.
* Zeus dengan Bertha, namun karena Bertha sudah dengan Xavier dan lebih memilih Xavier maka Zeus tidak mendapatkan pasangan.
* Victor dengan Amy, karena Amy lebih menyukai Victor sehingga Yancey dilepas.
* Yancey dengan Diane, karena Diane lebih menyukai Yancey sehingga Wyatt dilepas.
* Zeus dengan Diane, karena Diane lebih memilih Zeus sehingga Yancey dilepas.
* Wyatt dengan Clare.
* Yancey dengan Erika.

Hasil akhir matchmaking:

* Victor dengan Amy
* Xavier dengan Bertha
* Wyatt dengan Clare
* Zeus dengan Diane
* Yancey dengan Erika

Pertanyaan:

* 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:
  + 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 n2 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!

**Jawab**

Program untuk pseudocode di atas:

#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;

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;

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

    }

    return false;

}

void check\_stability(const Couples &engaged, const PrefMap &men\_pref, const PrefMap &women\_pref){

    cout << "Stablility:\n";

    bool stable = true;

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

        const string &bride = it->first;

        const string &groom = it->second;

        const PrefList &preflist = men\_pref.at(groom);

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

            if (\*it == bride)

                break;

            if (prefers(preflist, \*it, bride) && prefers(women\_pref.at(\*it), groom, engaged.at(\*it))){

                cout << "\t" << \*it <<

                    " prefers " << groom <<

                    " over " << engaged.at(\*it) <<

                    " and " << groom <<

                    " prefers " << \*it <<

                    " over " << bride << "\n";

                stable = false;

            }

        }

    }

    if (stable) cout << "\t(all marriages stable)\n";

}

int main(){

    PrefMap men\_pref, women\_pref;

    queue<string> bachelors;

    for (int i = 0; i < 5; ++i){

        for (int j = 1; j < 6; ++j){

        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;

    cout << "Matchmaking:\n";

    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()){

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

                engaged[bride] = suitor;

                break;

            }

            const string &groom = engaged[bride];

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

                cout << "\t" << bride << " dumped " << groom << " for " << suitor << "\n";

                bachelors.push(groom);

                engaged[bride] = suitor;

                break;

            }

        }

        bachelors.pop();

    }

    cout << "Engagements:\n";

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

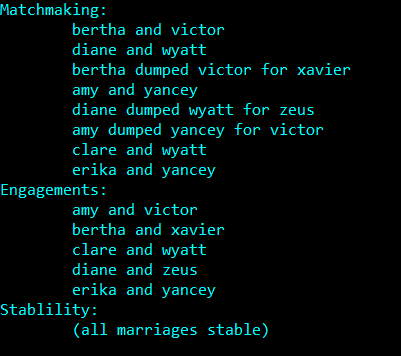
        cout << "\t" << it->first << " and " << it->second << "\n";

    }

    check\_stability(engaged, men\_pref, women\_pref);

}

Output dari program:



**Analisis:**

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

Ya, jawabannya sama persis.

* 1. 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 n2 iterasi menggunakan While Loop. Buktikan!

🡪 Setiap laki-laki akan memilih satu wanita yang belum pernah ia pilih sebelumnya. Karena ada n laki-laki dan setiap laki-laki memiliki preferensi sebanyak n, ada banyak kemungkinan besar proposal yang dapat terjadi. Jumlah iterasi yang dapat terjadi paling banyak adalah n2 karena sebanyak n pria akan melamar minimal sekali n preferensi wanita sampai terjadi kecocokan.

* 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.

* 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 urut preferensi. Dengan itu Himpunan S adalah perfect matching dikarenakan teori tersebut.

* 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.