TUGAS 1 PRAKTIKUM

STABLE MATCHING PROBLEM



Alvin

140810180013

PROGRAM STUDI S-1 TEKNIK INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS PADJADJARAN

2020

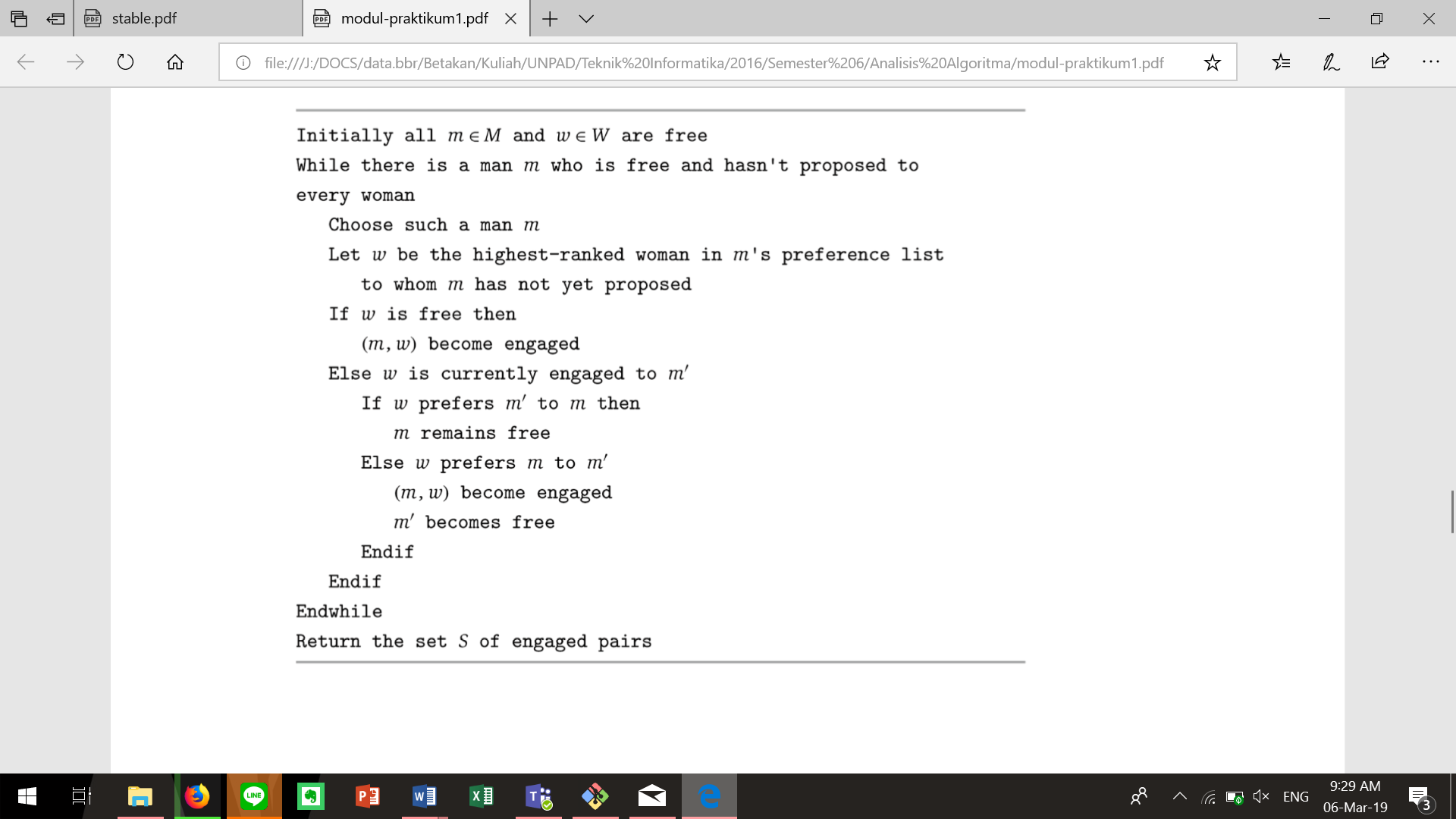
**Soal**



Hasil yang diharapkan:



Dengan pseudocodenya sebagai berikut:



Pertanyaan:

1. Ubahlah pseudocode algoritma G-S pada worksheet 01 ke dalam program menggunakan bahasa C++.
2. Gunakan table pria sebagai table acuan untuk memudahkan Anda menentukan pasangannya.
3. Cocokkan jawaban Anda pada worksheet 01 dengan hasil program yang Anda buat.
4. Jika ada yang berbeda tuliskan bagian mana yang berbeda dan analisalah:
   1. Apakah jawaban Anda di Worksheet 01 dan Program sama persis? Jika Tidak? Kenapa?
   2. 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!

**Jawaban**

Men’s Preferences Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0th | 1st | 2nd | 3rd | 4th |
| Bertha | Amy | Diane | Erika | Clare |
| Diane | Bertha | Amy | Clare | Erika |
| Bertha | Erika | Clare | Diane | Amy |
| Amy | Diane | Clare | Bertha | Erika |
| Bertha | Diane | Amy | Erika | Clare |

Victor

Wyatt

Xavier

Yancey

Zeus

Women’s Preferences Profile

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0th | 1st | 2nd | 3rd | 4th |
| Zeus | Victor | Wyatt | Yancey | Xavier |
| Xavier | Wyatt | Yancey | Victor | Zeus |
| Wyatt | Xavier | Yancey | Zeus | Victor |
| Victor | Zeus | Yancey | Xavier | Wyatt |
| Yancey | Wyatt | Zeus | Xavier | Victor |

Amy

Bertha

Clare

Diane

Erika

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.

Pasangan yang terbentuk adalah

* + Victor – Amy
  + Wyatt - Clare
  + Xavier – Bertha
  + Yancey - Erika
  + Zeus – Diane

Program untuk pseudocode di atas:

*/\**

*Nama : Alvin*

*NPM : 140810180013*

*Kelas : A*

*Tugas : Stable Matching Problem*

*\*/*

#include <iostream>

#include <string.h>

#include <stdio.h>

**using** **namespace** std**;**

#define N 5

**bool** pilihPria**(int** prioritas**[**2 **\*** N**][**N**],** **int** w**,** **int** m**,** **int** m1**){***//mengecek prioritas*

**for** **(int** i **=** 0**;** i **<** N**;** i**++)**

**{**

**if** **(**prioritas**[**w**][**i**]** **==** m1**)**

**return** **true;**

**if** **(**prioritas**[**w**][**i**]** **==** m**)**

**return** **false;**

**}**

**}**

**void** pencocokan**(int** prioritas**[**2 **\*** N**][**N**])** **{**

**int** pasanganWanita**[**N**];**

**bool** priaSingle**[**N**];**

memset**(**pasanganWanita**,** **-**1**,** **sizeof(**pasanganWanita**));**

memset**(**priaSingle**,** **false,** **sizeof(**priaSingle**));**

**int** jumlahSingle **=** N**;**

**while** **(**jumlahSingle **>** 0**){**

**int** m**;***//inisialisasi pasangan pria*

**for** **(**m **=** 0**;** m **<** N**;** m**++)**

**if** **(**priaSingle**[**m**]** **==** **false)**

**break;**

**for** **(int** i **=** 0**;** i **<** N **&&** priaSingle**[**m**]** **==** **false;** i**++){***//pria single memilih pasangan berdasarkan prioritas tertingginya*

**int** w **=** prioritas**[**m**][**i**];**

**if** **(**pasanganWanita**[**w **-** N**]** **==** **-**1**){***//jika wanita belum bertunangan*

pasanganWanita**[**w **-** N**]** **=** m**;**

priaSingle**[**m**]** **=** **true;**

jumlahSingle**--;**

**}**

**else{**

**int** m1 **=** pasanganWanita**[**w **-** N**];***//jika wanita sudah bertunangan*

**if** **(**pilihPria**(**prioritas**,** w**,** m**,** m1**)** **==** **false){**

pasanganWanita**[**w **-** N**]** **=** m**;**

priaSingle**[**m**]** **=** **true;**

priaSingle**[**m1**]** **=** **false;**

**}**

**}**

**}**

**}**

cout **<<** **" HASIL PASANGAN \n"** **<<** endl**;**

string pria**;**

string wanita**;**

**for** **(int** i **=** 0**;** i **<** N**;** i**++){**

**if** **(**i **<** N**){**

**if** **(**pasanganWanita**[**i**]** **==** 0**)**

pria **=** **"Victor";**

**if** **(**pasanganWanita**[**i**]** **==** 1**)**

pria **=** **"Wyatt";**

**if** **(**pasanganWanita**[**i**]** **==** 2**)**

pria **=** **"Xavier";**

**if** **(**pasanganWanita**[**i**]** **==** 3**)**

pria **=** **"Yancey";**

**if** **(**pasanganWanita**[**i**]** **==** 4**)**

pria **=** **"Zeus";**

**if** **(**i **==** 0**)**

wanita **=** **"Amy";**

**if** **(**i **==** 1**)**

wanita **=** **"Bertha";**

**if** **(**i **==** 2**)**

wanita **=** **"Clare";**

**if** **(**i **==** 3**)**

wanita **=** **"Diane";**

**if** **(**i **==** 4**)**

wanita **=** **"Erika";**

**}**

cout **<<**pria **<<** **"\t& "** **<<** wanita **<<** endl**;**

**}**

**}**

**int** main**()**

**{**

**int** prioritas**[**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**}**

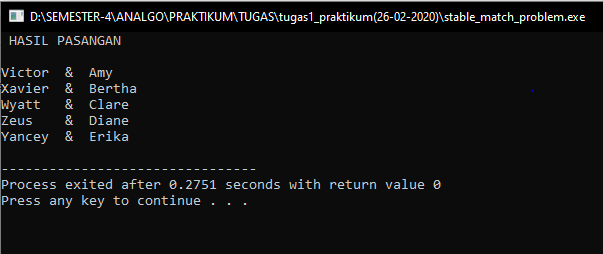
**};**

pencocokan**(**prioritas**);**

**return** 0**;**

**}**

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 pria yang ada akan memilih satu wanita yang sangat disukainya berdasarkan table preferensi yang ada, jika terdapat n pria dan n wanita maka agar semua mendapatkan pasangan maka terjadi n2 loop hingga semua berpasangan..

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

🡪 dalam pelaksanaan algoritma G-S setiap pria akan melamar wanita yang berbeda, jika semua pasangan sudah bertunangan maka tidak mungkin ada seorang wanita yang belum pernah diajak bertunangan sama sekali.

* 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.