

**LAPORAN PRAKTIKUM ANALISIS**

**ALGORITMA**

**JUDUL :**

**Stable Matching Problem (SMP)**

**Dikerjakan Oleh:**

**Kevin Andrew 140810160012 2016**

**UNIVERSITAS PADJADJARAN**

**SUMEDANG**

**2018**

**WORKSHEET 01**

1. m = victor

victor 🡪 Bertha

if (Bertha == free) //True

**(Victor, Bertha)**

1. m = wyatt

wyatt 🡪 Diane

if (Diane == free) //true

**(wyatt, diane)**

1. m = Xavier

Xavier 🡪 Bertha

If (bertha == free) //false

Else

If (bertha prefer victor) //false

Else (bertha prefer Xavier) //true

**(Xavier, bertha)**

**Victor free**

1. m = Yancey

Yancey 🡪 Amy

If (Amy == free) //true

**(Yancey, Amy)**

1. m = Zeus

Zeus 🡪 Bertha

If (Bertha == free) //false

Else

If (bertha prefer Xavier) // true

**(Xavier, Bertha)**

**Zeus free**

1. m = Victor

Victor 🡪 Amy

If (Amy == free) //false

Else

If (amy prefer yancey) //false

Else (amy prefer victor) //true

**(Victor, Amy)**

**Yancey free**

1. m = Zeus

Zeus 🡪 Diane

If (Diane == free) //false

Else

If (Diane prefer wyatt) //false

Else (Diane prefer zeus) //true

**(Zeus, Diane)**

**Wyatt free**

1. m = Yancey

yancey 🡪 Diane

if (diane == free) //false

else

if (diane prefer zeus) //true

**(Zeus, Diane)**

**Yancey free**

1. m = wyatt

wyatt 🡪 bertha

if (bertha == free) // false

else

if (bertha prefer Xavier) //true

**(Xavier, Bertha)**

**Wyatt free**

1. m = yancey

yancey 🡪 clare

if (clare == free) //true

**(Yancey, Clare)**

1. m = wyatt

wyatt 🡪 Amy

if (amy == free) //false

else

if (amy prefer victor) //true

**(Victor, Amy)**

**Wyatt free**

1. m = wyatt

wyatt 🡪 Clare

if (clare == free) //false

else

if (clare prefer yancey) //false

else (clare prefer wyatt) //true

**(Wyatt, Clare)**

**Yancey free**

1. m = yancey

yancey 🡪 Erika

if (Erika == free) //true

**(Yancey, Erika)**

Jadi pasangannya :

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

**UBAHLAH PSEUDOCODE ALGORITMA G-S PADA WORKSHEET 01 KE DALAM PROGRAM MENGGUNAKAN BAHASA C++**

// C++ program for stable marriage problem

#include <iostream>

#include <string.h>

#include <stdio.h>

using namespace std;

// Number of Men or Women

#define N 5

// This function returns true if woman 'w' prefers man 'm1' over man 'm'

bool wPrefersM1OverM(int prefer[2\*N][N], int w, int m, int m1)

{

// Check if w prefers m over her current engagment m1

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

{

// If m1 comes before m in lisr of w, then w prefers her

// cirrent engagement, don't do anything

if (prefer[w][i] == m1)

return true;

// If m cmes before m1 in w's list, then free her current

// engagement and engage her with m

if (prefer[w][i] == m)

return false;

}

}

// Prints stable matching for N boys and N girls. Boys are numbered as 0 to

// N-1. Girls are numbereed as N to 2N-1.

void stableMarriage(int prefer[2\*N][N])

{

// Stores partner of women. This is our output array that

// stores paing information. The value of wPartner[i]

// indicates the partner assigned to woman N+i. Note that

// the woman numbers between N and 2\*N-1. The value -1

// indicates that (N+i)'th woman is free

int wPartner[N];

// An array to store availability of men. If mFree[i] is

// false, then man 'i' is free, otherwise engaged.

bool mFree[N];

// Initialize all men and women as free

memset(wPartner, -1, sizeof(wPartner));

memset(mFree, false, sizeof(mFree));

int freeCount = N;

// While there are free men

while (freeCount > 0)

{

// Pick the first free man (we could pick any)

int m;

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

if (mFree[m] == false)

break;

// One by one go to all women according to m's preferences.

// Here m is the picked free man

for (int i = 0; i < N && mFree[m] == false; i++)

{

int w = prefer[m][i];

// The woman of preference is free, w and m become

// partners (Note that the partnership maybe changed

// later). So we can say they are engaged not married

if (wPartner[w-N] == -1)

{

wPartner[w-N] = m;

mFree[m] = true;

freeCount--;

}

else // If w is not free

{

// Find current engagement of w

int m1 = wPartner[w-N];

// If w prefers m over her current engagement m1,

// then break the engagement between w and m1 and

// engage m with w.

if (wPrefersM1OverM(prefer, w, m, m1) == false)

{

wPartner[w-N] = m;

mFree[m] = true;

mFree[m1] = false;

}

} // End of Else

} // End of the for loop that goes to all women in m's list

} // End of main while loop

// Print the solution

cout << "Woman Man" << endl;

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

cout << " " << i+N << "\t" << wPartner[i] << endl;

}

// Driver program to test above functions

int main()

{

/\*

0 victor, 1 wyat, 2 xavier, 3 yancey, 4 zeus

5 amy, 6 bertha, 7 clare, 8 diane, 9 erika

\*/

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}

};

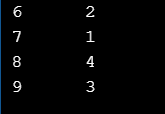
stableMarriage(prefer);

return 0;

}

OUTPUT :

Keterangan

0 victor

1 wyat

2 xavier

3 yancey

4 zeus

5 amy

6 bertha

7 clare

8 diane

9 erika

Analisis Algoritma

Jawablah pertanyaan berikut:

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

Jawab : Iya sama

1. Algoritma G-S berakhir setelah paling banyak n2 iterasi menggunakan While Loop.

Buktikan!

Jawab : Karna kemungkinan paling banyak adalah 5 (jumlah cowo) x 5 (jumlah cewe) yaitu 25

Kemungkinan. Di setiap iterasi sementara loop, seorang pria lajang melamar wanita berikutnya dalam daftar pilihannya, seseorang yang tidak pernah dia miliki diusulkan sebelumnya. Karena ada n laki-laki dan setiap daftar preferensi memiliki panjang n, ada paling banyak n2 proposal yang bisa terjadi. Jadi jumlah iterasi yang dapat terjadi paling banyak adalah n2

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

Jawab : Buktinya dengan kontradiksi. Misalkan ada waktu tertentu dalam pelaksanaan

algoritma ketika seorang pria lajang, namun telah diusulkan untuk setiap wanita. Ini

artinya dengan ini Waktu, setiap wanita telah diusulkan setidaknya satu kali. Dengan

Lemma 1, kami mendapatkan itu setiap wanita bertunangan. Dengan demikian, kami telah n melibatkan perempuan dan karenanya n bertunangan, yang menyiratkan bahwa m adalah juga terlibat bertentangan dengan asumsi kami bahwa m adalah lajang.

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

Jawab : Buktinya dengan kontradiksi. Misalkan tidak, maka ada seorang pria yang masih lajang di akhir algoritma. Oleh Lemma 2, itu berarti m belum melamar beberapa wanita. Tetapi kemudian, algoritma tidak akan keluar dari sambil memutar, menghasilkan kontradiksi yang diinginkan.

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

Jumlah :

Sekarang kami menunjukkan bahwa pencocokan yang dikembalikan stabil. Lagi-lagi buktinya dengan kontradiksi. Seharusnya ada laki-laki m dan m0 dan perempuan b dan b

0 sedemikian rupa sehingga (m, w) dan (m0, w0) berada di S, tetapi m lebih suka w0 ke w dan w0 lebih suka m ke m0

.