GeeksforGeeks
A computer science portal for geeks

GeeksQuiz

Login

Home Algorithms DS GATE Interview Corner Q&A C C++ Java Books Contribute Ask a Q About

Array Bit Magic C/C++ Articles GFacts Linked List MCQ Misc Output String Tree Graph

Stable Marriage Problem

Given N men and N women, where each person has ranked all members of the opposite sex in order of preference, marry the men and women together such that there are no two people of opposite sex who would both rather have each other than their current partners. If there are no such people, all the marriages are "stable" (Source Wiki).

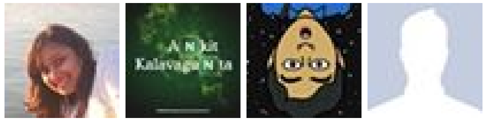
Consider the following example.

Let there be two men m1 and m2 and two women w1 and w2.
Let m1's list of preferences be {w1, w2}
Let m2's list of preferences be {w1, w2}
Let w1's list of preferences be {m1, m2}
Let w2's list of preferences be {m1, m2}

The matching { {m1, w2}, {w1, m2} } is not stable because m1 and w1 would prefer each other over their assigned partners. The matching {m1, w1} and {m2, w2} is stable because there are no two people of opposite sex that would prefer each other over their assigned partners.

It is always possible to form stable marriages from lists of preferences (See references for proof).
Following is Gale–Shapley algorithm to find a stable matching:
The idea is to iterate through all free men while there is any free man available. Every free man goes to all women in his preference list according to the order. For every woman he goes to, he checks if the woman is free, if yes, they both become engaged. If the woman is not free, then the woman chooses either says no to him or dumps her current engagement according to her preference list. So an engagement done once can be broken if a woman gets better option.
Following is complete algorithm from Wiki

```
Initialize all men and women to free
```



Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

```
while there exist a free man m who still has a woman w to propose to
{
    w = m's highest ranked such woman to whom he has not yet proposed
    if w is free
       (m, w) become engaged
    else some pair (m', w) already exists
       if w prefers m to m'
          (m, w) become engaged
           m' becomes free
       else
          (m', w) remain engaged
}
```

Input & Output: Input is a 2D matrix of size (2*N)*N where N is number of women or men. Rows from 0 to N-1 represent preference lists of men and rows from N to 2*N – 1 represent preference lists of women. So men are numbered from 0 to N-1 and women are numbered from N to 2*N – 1. The output is list of married pairs.

Following is C++ implementation of the above algorithm.

```
// C++ program for stable marriage problem
#include <iostream>
#include <string.h>
#include <stdio.h>
using namespace std;

// Number of Men or Women
#define N  4

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

Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding "extern" keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.
- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

```
        // 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
// 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()
{
    int prefer[2*N][N] = { {7, 5, 6, 4},
        {5, 4, 6, 7},
        {4, 5, 6, 7},
        {4, 5, 6, 7},
        {0, 1, 2, 3},
        {0, 1, 2, 3},
        {0, 1, 2, 3},
        {0, 1, 2, 3},
    };
    stableMarriage(prefer);

    return 0;
```

705 Subscribe

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 4 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 8 minutes ago

Sanjay Agarwal bool tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 32 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 34 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 59 minutes ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

```
}
```

Output:

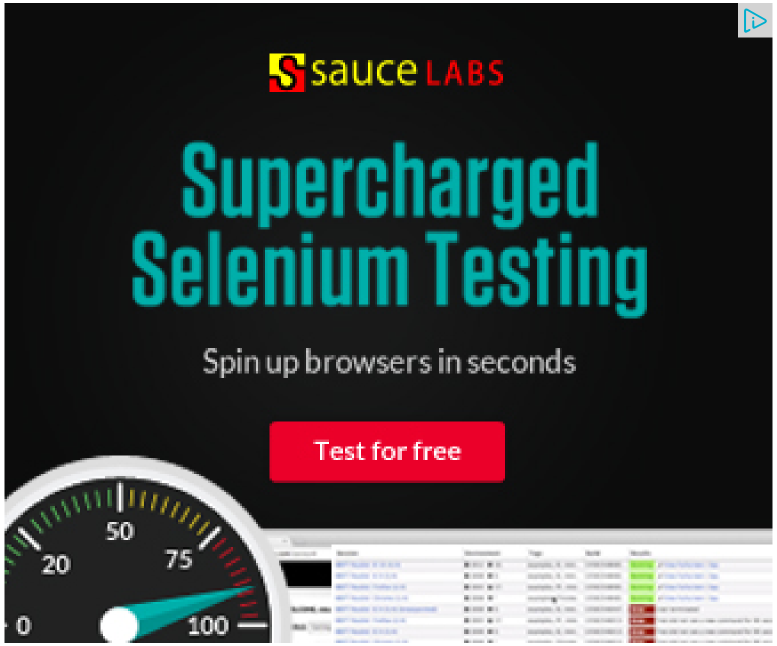
```
Girl    Boy
4        2
5        1
6        3
7        0
```

References:

http://www.csee.wvu.edu/~ksmani/courses/fa01/random/lecnotes/lecture5.pdf

http://www.youtube.com/watch?v=5RSMLgy06Ew#t=11m4s

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



Related Tpoics:

- Remove minimum elements from either side such that 2*min becomes more than max

- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3

45 Tweet 5 0

Writing code in comment? Please use **ideone.com** and share the link here.

29 Comments **GeeksforGeeks**

Sort by Newest ▾

Join the discussion…

divyansh8063 • 2 months ago

Using STACK --

```
main()
{
int n,i,j,z,tmp,f1;
cin>>n;
int a[2*n][n],b[2*n+1];
stack<int>my;
REP(i,2*n)
{
b[i]=-1;
REP(j,n)
cin>>a[i][j];
}
REP(i,n) my.push(i);
```

```
while(!my.empty())
{
tmp=my.top();
my.pop();
```

see more

⌃ | ⌄ • Reply • Share ›

Ankit Chaudhary • 4 months ago

I think worst case complexity of above algorithm is O(n^2).
This will occur when in first iteration out of n men only one remain engaged.
2nd iteration : out of n-1 only 1 men remain engaged.
3rd iteration : out of n-2 only 1 men remain engaged.
and so on.
Eg :
men list :
0,1,2,3
0,1,2,3
0,1,2,3
0,1,2,3

women list :
3,2,1,0
3,2,1,0
3,2,1,0
3,2,1,0

Best Case : O(n)

see more

⌃ | ⌄ • Reply • Share ›

Ankit Chaudhary ➦ Ankit Chaudhary • 4 months ago

Optimization :

1 : As said by skulldude below.

2 : Instead of using iteration from 0 to n-1 to find free men, use queue w
remain same, but it is better then above.
if men is free insert it into queue.
If engaged remove it from queue.
Initially fill queue will all men
So while condition will be :

```
while(queue not empty)
{

}
```

^ | ⌄ • Reply • Share ›

zzer ➦ Ankit Chaudhary • a month ago

nice, this is the optimized code below:

```
bool woman_prefer(int prefer[2*N][N],int woman,int man1,int m

{

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

{

//if we encounter man1 first, which indicates man1 ranks highe

if(prefer[woman][i] == man1)

return true;

if(prefer[woman][i] == man2)

return false;
```

see more

^ | ˅ • Reply • Share ›

Guest • 6 months ago

For Python Lovers:

```
from sys import stdin
from cStringIO import StringIO

def main():
    inp = iter(StringIO(stdin.read())).next
    t = int(inp())
    for _ in xrange(t):
        n = int(inp())
        w = {}
        m = {}
        freem = set([`i+1` for i in xrange(n)])
        freew = set([`i+1` for i in xrange(n)])
        mproposed = {}
        engaged = {}
        for i in xrange(n):
            w[`i+1`] = inp().split()[1:]
```

see more

^ | ˅ • Reply • Share ›

Math • 6 months ago

Is there any example for Stable marriage problem in Almms ? I have a problen one but in Almms and I never worked with this plateform. Tks

^ | ˅ • Reply • Share ›

ajay • 7 months ago

for best case say each man has different preference for woma then what will l
case where each man likes same woman the most????/ Please tell as fast a

⌃ | ⌄ • Reply • Share ›

mrn • 8 months ago

bipartite problem ...

⌃ | ⌄ • Reply • Share ›

Abhishek J • 9 months ago

what will be the time complexity of this algorithm ?
i think it will be depending on priority factor !

⌃ | ⌄ • Reply • Share ›

Abhishek J • 9 months ago

what will be the time complexity of this algorithm ?
i think it will be depending on the preference list?

⌃ | ⌄ • Reply • Share ›

coolabhi • 9 months ago

Shouldn't the function name wprefersm1overm be named as wprefersmovern
checking if
w prefers m over her current engagment m1

```
/* Paste your code here (You may delete these lines if not writing c
```

⌃ | ⌄ • Reply • Share ›

skulldude • 9 months ago

I think we can add one optimization to the above code. Instead of checking thr
for the current free man, we can remember every man's last proposal(in an a

This is true because, once a woman rejects a man M, she is never going to ac
had a better man when she rejected M. So, there is no use for a man to propo
before and had been rejected.

Please comment if you find anything wrong in it.

Thanks-
Balasubramanian

Reply • Share ›

zzer skulldude • a month ago

the pseudo-code " w = m's highest ranked such woman to whom he h
this idea

Reply • Share ›

Ankit Chaudhary skulldude • 4 months ago

Agree with u

Reply • Share ›

Atul • 10 months ago

```
     // An array to store availability of men.  If mFree[i] is
    // false, then man 'i' is free, otherwise engaged.
```

Wouldn't it be logical to change the name from "mFree" to "mEngaged". It wou
logic.

1 • Reply • Share ›

Abhishek Bussa • 10 months ago

Code in C#:

https://github.com/phoenix2fir...

Reply • Share ›

Verma Shailendra • 10 months ago

but the solution may differ if we start with woman,, I mean every free woman is
preference list.

Reply • Share ›

Vishal Hemnani ➦ Verma Shailendra • 8 months ago

Yes. The stable solution will be different when we start it with women p
algorithm is men-biased.

Reply • Share ›

Abhishek • 10 months ago

In

```
for (m = 0; m < N; m++)
           if (mFree[m] == false)
               break;
```

shouldn't the condition be if(mFree[m]==True) ?

Reply • Share ›

kartik ➦ Abhishek • 10 months ago

I think the condition is fine. mFree[m] false means that the man m is fr

Reply • Share ›

Abhishek ➦ kartik • 10 months ago

// An array to store availability of men. If mFree[i] is true, then m
bool mFree[N];

Then i think we should change the comment, as comment says

Reply • Share ›

GeeksforGeeks ➦ Abhishek • 10 months ago

Thanks Abhishek, we have updated the comment.

^ | v • Reply • Share ›

Atul ➦ GeeksforGeeks • 10 months ago

>```
>       // An array to store availability of men.  I
>     // false, then man 'i' is free, otherwise eng
>```

Wouldn't it be logical to change the name from "mFree"
it easier to understand the logic.

^ | v • Reply • Share ›

Rahul Singh (selfcompiler) • 10 months ago

```
//i use priority queue //
#include
#include
#include
#include
using namespace std;
#define FREE -1
class cmp
{
public:
bool operator()(pair a,pair b)
{
if(a.first<b.first)
return false;
else
return true;
}
```

};

see more

• Reply • Share ›

(GeeksFollower) • 10 months ago

correct me admin..
if there are N boys n girls..
There will be at max N*N engagement..
Now say for every pair, we calculate sum of their preferences for each other.
sort (in ascending order) these pairs based on this calculated sum..
Now marry the pairs based on this preference..

I tried for so many test cases.. it worked.. ur better analysis would be helpful.

• Reply • Share ›

Vibhu Tiwari • 10 months ago

@GeeksforGeeks:
I think that in the if condition of wprefersm1overm wont the wPartner[w-N]=m a
N] instead of m1.

• Reply • Share ›

GeeksforGeeks ➜ Vibhu Tiwari • 10 months ago

Thanks for pointing this out. We have updated the line.

• Reply • Share ›

vamshi ➜ GeeksforGeeks • 5 months ago

why did we use wPartner[w-N] in all the places...why didn't we

• Reply • Share ›

Neha ➜ vamshi • 4 months ago

What is sPartner[w]

Subscribe Add Disqus to your site