Comp 496ALG Project 1 - Stable Matching Problem

Points: 35

Due: Sept 29

1. Implement the Brute Force Stable Matching Algorithm as follows: Read n and male and female preferences from a file, using INPUT FILE SETUP in item 3. Generate all possible permutations of 0,1,2, …n-1. Each permutation corresponds to a matching of n men and n women. Check for instabilities in each matching. Print out the matchings that have no instabilities. Test for correctness. [HINT: You may use the web or a book to find an algorithm that nicely generates all permutations of 1, 2,..,n. Adapt so that it gives permutations of 0 to n-1. Give source in your report. ]
2. Implement the G-S Stable Matching algorithm studied in class (and in text). Read n and male and female preferences from a file, using INPUT FILE SETUP in item 3. Print out the stable matching that the algorithm finds. Test for correctness. Compare with results from item 1.
3. INPUT FILE SETUP: Number the males 0,…, n-1 and the females 0,1,…n-1

The input file for male and female preferences (just the numbers) will look like this:

6 <- n

1 3 4 2 5 0 <- male preferences. For example male 0 prefers women in order 1 3 4 2 5 0

5 4 3 2 1 0

1 3 4 2 5 0

1 2 3 0 5 4

0 5 4 2 1 3

0 4 3 5 2 1 <- male preferences. For example male 5 prefers women in order 0 4 3 5 2 1

1 3 5 0 4 2 <- female preferences. For example female 0 prefers men in order 1 3 5 0 4 2

4 0 3 5 2 1

2 4 3 1 5 0

0 4 2 3 5 1

1 3 2 4 0 5

1 4 0 3 2 5 <- female preferences. For example female 5 prefers men in order 1 4 0 3 2 5

1. Run time analysis of Brute Force algorithm. Run algorithm for n = 6, 8,10,11,12. Make up “random” test cases. Track how much time (msecs) it takes to run each case. Let T(n) = time in msecs to run your algorithm for n men and n women. Graph the T(n) as a function of n. Predict T(13) and T(14).
2. Analyze your Brute Force algorithm to find T(n) as a Big O of some function. Show work.
3. Run time analysis of G-S Stable Matching algorithm. Run algorithm for n = n = 5, 10, 15, 50. Make up “random” test cases. Track how much time (msecs) it takes to run each case. Let T(n) = time in msecs to run your algorithm for n men and n women. Graph the T(n) as a function of n. Predict T(100).
4. Analyze your G-S Stable Matching algorithm to find T(n) as a Big O of some function. Show work.
5. Instructor Test Cases: Posted on Sept 22. These will be text files.
6. HAND IN: (1) your source code; (2) runs of instructor’s test cases; (3) detailed typewritten report on items 4 - 7, in order. Use Excel or something similar to plot your graphs. Put a cover page on your work.

**Sample Test Case 1**

3  
1 2 0  
0 1 2  
1 2 0

1 0 2  
1 2 0  
0 1 2

Stable Matches from Brute Force Algorithm – Prints male female pairs

[(0,2), (1,0), (2 ,1) ]

**Sample Test Case 2**

8  
5 0 1 6 4 7 2 3   
4 0 2 3 7 6 1 5   
7 5 6 0 4 1 2 3   
0 7 4 6 5 1 3 2   
1 5 7 3 2 0 4 6   
2 3 6 5 7 1 0 4   
4 0 2 7 5 6 3 1   
0 3 1 4 6 2 5 7  
   
4 5 2 0 7 1 6 3   
6 1 7 0 5 2 3 4   
1 3 0 7 4 5 2 6   
6 7 5 3 1 2 4 0   
1 3 4 6 5 7 0 2   
1 2 6 7 5 4 0 3   
1 3 4 7 6 2 0 5   
4 6 3 2 1 7 5 0

Stable Matches from Brute Force algorithm. Prints male female pairs.

[ (0,0) , (1,4) , (2,5) , (3,6) , (4,1) , (5,2) , (6,7) , (7,3) ] 🡨 G-S Stable Matching Algorithm gives this one

[ (0,0) , (1,4) , (2,5) , (3,6) , (4,7) , (5,2) , (6,3) , (7,1) ]

[ (0,2) , (1,4) , (2,5) , (3,6) , (4,7) , (5,0) , (6,3) , (7,1) ]