

# COMPSCI 320SC 2021 Midterm Test

University ID: \_\_\_\_\_

Student Name: \_\_\_\_\_

Student Signature: \_\_\_\_\_

Time Finished: \_\_\_\_\_

Attempt *all* questions by yourself.

Write clearly and *show all your work!*

Marks for each question are shown after each part of the questions.

This 60 minute test is worth 10% of your final grade for the course.

An extra 30 minutes for typesetting/scanning is added to Canvas for uploading PDF answers.

Make sure your Name and UPI is clearly displayed on your submitted PDF. (This is inside the rendered document, not the filename!)

**Good luck!**

Question #:	1	2	3	Total
<i>Possible marks:</i>	10	10	10	30
<i>Awarded marks:</i>				

1. Stable matching problem of  $n$  men and  $n$  women

[10 marks]

- (a) As similar to the setting of Assignment 1, we consider odd integers as man ID and even integers as woman ID. Consider the following tables as preferences of men and women. What is the man-optimal stable matching provided by the Gale-Shapley algorithm? How many proposals does the algorithm use?

Table 1: Men's preferences

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

Table 2: Women's preferences

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

(4 marks)

- (b) To find a stable matching, the Gale-Shapley algorithm has to implement the method  $prefer(w, m, m')$  that checks whether a woman  $w$  prefers a man  $m$  to the current partner  $m'$ . A simple way of implementing  $prefer(w, m, m')$  is called  $index(w, m, m')$  that scans the preference of  $w$  to find the index of  $m$  and  $m'$ . If  $m$  appears earlier in the list than  $m'$ , return True; otherwise, return False.

It is easy to see that  $index(w, m, m')$  runs in  $O(n)$  time. It is suggested that for every woman  $w$  we create the inverse of preference list of men to speed up the Gale-Shapley algorithm. However, creating the inverse of a preference list takes  $O(n)$  time, which is similar to  $index(w, m, m')$ .

Explain why the trick of using inverse of preference list for each woman can improve the running time of Gale-Shapley algorithm. You will get no marks for writing down a big-Oh expression with no explanation.

(2 marks)

- (c) Given a setting of the stable matching problem where there is a man  $m$  who is the first choice of *all* women. Prove or give a counterexample: In any stable matching,  $m$  must be matched with his first choice.

(4 marks)

## 2. Greedy Algorithms

[10 marks]

Consider the undirected and weighted graph in Figure 1 below. On the undirected graph, the graph can be traversed in either directions. For example, you can traverse from Node 1 to Node 2 and from Node 2 to Node 1 with the same weight of 3.

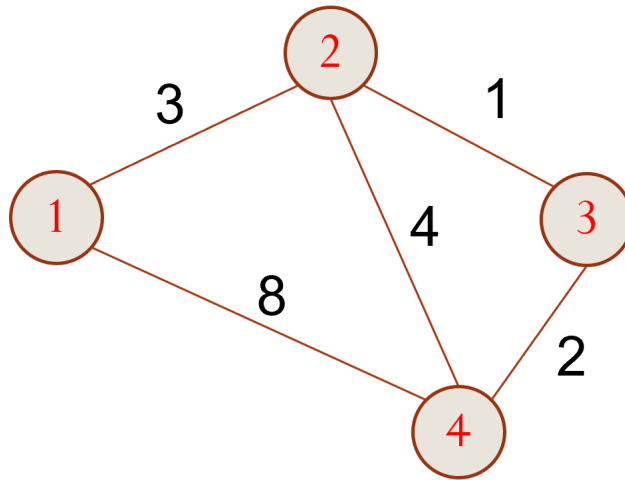


Figure 1: Graph 1.

- Draw the shortest path from Node 1 to Node 4 using the Dijkstra's algorithm on Graph 1. List the order in which nodes are added to the set of explored nodes. (2 marks)
- Draw the minimum spanning tree (MST) using the Prim's algorithm on Graph 1, starting at Node 1. List the order in which edges are added to the tree. (2 marks)
- On Graph 1, we observe that we can find the shortest path from any node to another node using only the MST. Is this observation correct for any undirected and weighted graph? Explain your answer. You will get no marks without any explanation.

(6 marks)

## 3. Divide and Conquer

[10 marks]

(a) Master theorem: Consider the following “divide-and-conquer” function:

```

function printer(int n)
  for i = 1 to n do
    for j = 1 to  $\lfloor \sqrt{n} \rfloor$  do
      print “Hello Amigo”
    end for
  end for
  if n > 0 then
    printer( $\lfloor 2n/3 \rfloor$ )
  end if

```

Let  $T(n)$  denote the number of lines generated by a call of  $printer(n)$ .

- i. Provide a recurrence equation for  $T(n)$ . (3 marks)
- ii. Solve the recurrence asymptotically for general  $n$ . (2 marks)

You may want to make use of the following *master recurrence theorem*: Assume  $T(n) = aT(n/b) + g(n)$ , where  $g(n)$  is  $\Theta(n^c)$ , is the total time for a divide and conquer algorithm. Then:

$$T(n) = \begin{cases} \Theta(n^c), & \text{if } a < b^c \\ \Theta(n^c \log n), & \text{if } a = b^c \\ \Theta(n^{\log_b a}), & \text{if } a > b^c \end{cases}$$

(b) Counting number of medians in a sorted array

A median item in an array (with duplicates) of  $n$  elements is the item  $x$  that is ranked  $\lfloor n/2 \rfloor$  in a sorted version of the array, where items are ranked 1 to  $n$ . Given a sorted array, we want to count the number of occurrences of the median in the array. Examples are given below.

Input: [3, 4, 4, 4, 5, 6, 7]

Output: 3

Input: [3, 3, 4, 4, 4, 4, 7]

Output: 4

Input: [3, 3, 3, 4, 6, 6, 6, 6]

Output: 1

Describe a divide-and-conquer algorithm that solves this problem in  $O(\log n)$  time where  $n$  is the size of the array. (5 marks)