

Name:

SID:

GSI and section time:

*Write down the names of the students on your left and right as they appear on their SID.*

Name of student on your left:

Name of student on your right:

Name of student behind you:

Name of student in front of you:

Note: For each question part, 20% of the points will be given to any blank answer, or to any *clearly* crossed-out answer.

Note: If you finish in the last 15 minutes, please remain seated and not leave early, to not distract your fellow classmates.

Instructions: You may consult one handwritten, single-sided sheet of notes. You may not consult other notes, textbooks, etc. Cell phones and other electronic devices are not permitted.

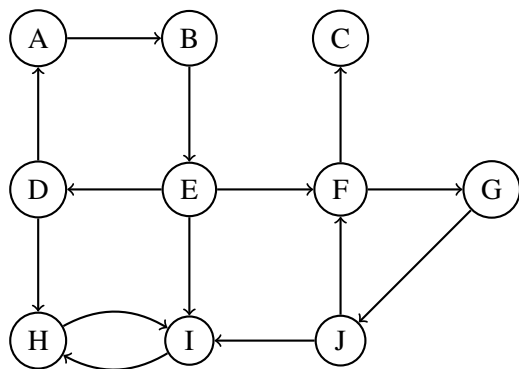
There are 8 questions. The last page is page 12.

Answer all questions. On questions asking for an algorithm, make sure to respond in the format we request. Be precise and concise. Write in the space provided. Good luck!

Do not turn this page until an instructor tells you to do so.
---

**1. (10 pts.) Strongly Connected Components**

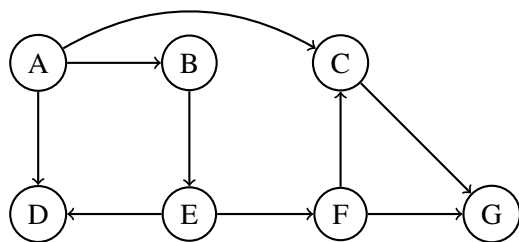
- (a) **(6 pts.)** For the directed graph below, find the strongly connected components and draw the DAG of the strongly connected components.



- (b) **(2 pts.)** List a set of edges which, if added to the graph, would make the graph have exactly one sink SCC and one source SCC. Use the fewest edges possible; if the graph already has one source SCC and one sink SCC, write “None.”
- (c) **(2 pts.)** List a set of edges which, if added to the graph, would make the whole graph strongly connected. Use the fewest edges possible; if the graph is already strongly connected, write “None.”

## 2. (15 pts.) Topological Sort

(a) (5 pts.) For the directed graph below, give a topological ordering of the vertices.



(b) (5 pts.) Now consider *all possible* topological sorts of this graph. For each following pair of vertices  $X$  and  $Y$ , write  $<$  if  $X$  must be before  $Y$  in any topological sort;  $>$  if  $X$  must be after  $Y$ ; and  $\leq$  if  $X$  can be either before or after  $Y$ .

C \_\_\_\_\_ E

B \_\_\_\_\_ D

D \_\_\_\_\_ G

(c) (5 pts.) State a concise sufficient and necessary condition for when two vertices in a DAG can be in either order in a topological sort. (In other words, either one of them can be first in a valid topological sort of that DAG).

**3. (20 pts.) True/False**

For each part, determine whether the statement is true or false. If true, prove the statement is true. If false, provide a simple counterexample demonstrating that it is false.

- (a) **(5 pts.)** The vertex with the smallest post-number in the DFS on a DAG is necessarily a sink vertex.

Mark one: TRUE or FALSE

- (b) **(5 pts.)** Arranging the vertices of a DAG according to increasing pre-number results in a topological sort.

Mark one: TRUE or FALSE

- (c) **(5 pts.)** Dijkstra's algorithm finds the shortest path even if some edge weights are 0 (and none are negative).

Mark one: TRUE or FALSE

- (d) **(5 pts.)** Recall the unique-shortest-paths question on Homework 4, where you are given an undirected graph  $G = (V, E)$  with edge lengths  $l_e > 0$ , starting vertex  $s \in V$  and asked to tell, for every vertex  $u$ , if there is a unique shortest path from  $s$  to  $u$ . This can be done by modifying Dijkstra's algorithm (lines 8-10 are added):

```
1: Initialize usp[.] to true
2: while  $H$  is not empty do
3:    $u = \text{DELETETEMIN}(H)$ 
4:   for all  $(u, v) \in E$  do
5:     if  $\text{dist}(v) > \text{dist}(u) + l(u, v)$  then
6:        $\text{dist}(v) = \text{dist}(u) + l(u, v)$ 
7:        $\text{DECREASEKEY}(H, v)$ 
8:        $\text{usp}[v] = \text{usp}[u]$ 
9:     else if  $\text{dist}(v) = \text{dist}(u) + l(u, v)$  then
10:       $\text{usp}[v] = \text{false}$ 
```

Does this algorithm still compute `usp` correctly if some edge weights are zero (and none are negative)?

Mark one: TRUE or FALSE

**4. (10 pts.) FFT Fundamentals**

(a) **(5 pts.)** What is the FFT of the vector  $[1, 1, 1]$ ?

(b) **(5 pts.)** Determine the polynomial of degree at most 3 that is represented by the following points:  
 $f(\omega^0) = 0, f(\omega^1) = 1, f(\omega^2) = 0, f(\omega^3) = 1$ . (Here  $\omega$  denotes the first of the 4th roots of unity.)

### 5. (15 pts.) Retroactive Oil Trading

The price of oil has been especially volatile over the last month. You want to figure out the largest profit you could have made in a single trade over that period, by buying oil at a certain time and selling at a later time. Specifically, given the chart of the oil price  $P[i]$  for  $i = 1$  to  $n$ , we're looking for the maximum value of  $(P[k] - P[j])/P[j]$  for some indices  $j, k$ , with  $1 \leq j < k \leq n$ .

- (a) (7 pts.) Fill in the blanks below to produce an algorithm that correctly solves this problem. You do not need to prove that your algorithm is correct.

OILMAX( $P[1..n]$ ):

1. If  $n \leq 1$ , return  $-\infty$ .
2. Set  $max_L := \text{OILMAX}(\text{_____})$ .
3. Set  $max_R := \text{OILMAX}(\text{_____})$ .
4. Set  $x := \text{_____}$ .
5. Set  $y := \text{_____}$ .
6. Return  $\max(\text{_____})$ .

- (b) (3 pts.) What is the runtime of the above algorithm? Give a short justification of your answer.

- (c) (5 pts.) Now, modify this algorithm to make it asymptotically faster. Clearly describe the changes you would make to the algorithm in (a) to speed it up, and state the new runtime. You do not need to justify the runtime or prove your new algorithm correct.

**6. (10 pts.) Arithmetic**

Given three subsets  $A, B, C$  of the set of integers  $\{1, \dots, n\}$ , determine which elements in  $C$  can be written as the sum of a pair of numbers from  $A$  and  $B$  (one each).

For example, if  $n = 6$ ,  $A = \{1, 3, 4\}$ ,  $B = \{3, 6\}$ , and  $C = \{2, 4, 5, 6\}$ , then the result would be  $\{4, 6\}$ .

There is a straightforward  $n^2$  solution; find a faster one.

*Please answer in the following format: clearly describe your algorithm (formal pseudocode is not necessary), justify why it is correct, and give and briefly justify the runtime.*



## 7. (15 pts.) Weighty Vertices

Consider an *undirected* graph  $G(V, E)$ , with nonnegative weights  $d(e)$  and  $w(v)$  for both edges and vertices. You have access to a Dijkstra's algorithm solver, which (in a standard undirected graph, with edge weights but no vertex weights) gives you the shortest path from a start vertex  $s$  to all other vertices.

Construct a graph which you can feed into the Dijkstra's solver to find the shortest path from the start vertex  $s \in V$  to all other vertices.

*Alternatively, for two-thirds credit, solve this problem assuming the edges are directed instead. If you choose this option, you have access to a Dijkstra's algorithm solver which works on directed graphs.*

*Please answer in the following format: clearly describe your graph, and prove that the path Dijkstra's returns on it will be the shortest path in the original graph.*

### 8. (20 pts.) Super Mario Path

You wish to use your knowledge of CS 170 to get a leg up on playing Super Mario Kart. In Level 17, you must drive through Bowser's Castle, which you can model as a directed graph. Traveling through each passage in the castle (directed edge) adds a positive or negative number of points to your score, depending upon what type of monster/object/power-up is on it. (Call  $w(e)$  to find the number of points (which can be negative) each edge adds.) To successfully traverse the level, you must find a path, *perhaps with repeated edges*, from the castle entrance  $s$  to the exit  $t$ , such that your score never goes negative *at any point in time*. Assume that every vertex is reachable from  $s$  and every vertex is able to reach  $t$ . Use your knowledge of CS 170 to design an efficient algorithm to find such a path if it exists, or to determine that there is no such path.

*Please answer in the following format: describe the main idea of your algorithm, write the pseudocode, justify why it is correct, and give and briefly justify the runtime.*

**Extra page**

*Use this page for scratch work, or for more space to continue your solution to a problem. If you want this page to be graded, please refer the graders here from the original problem page.*

**Extra page**

*Use this page for scratch work, or for more space to continue your solution to a problem. If you want this page to be graded, please refer the graders here from the original problem page.*