Midterm 1

Name:

SID:

Name of student to your left:

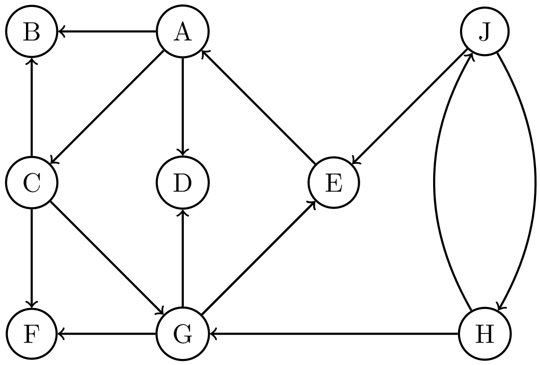
Name of student to your right:

GSI and section time:

Rules and Guidelines

- Answer all questions. Read them carefully first. Not all parts of a problem are weighted equally.
- Write your student ID number in the indicated area on each page.
- Be precise and concise.**Write in the solution box provided.** You may use the blank page on the back for scratch work, but it will not be graded. Box numerical final answers.
- Any algorithm covered in the lecture can be used as a blackbox.
- Throughout this exam (both in the questions and in your answers), we will use ω_n to denote the first n^{th} root of unity,i.e., $\omega_n = e^{2\pi i/n}$. So ω_{16} will denote the first 16^{th} root of unity, i.e., $\omega_{16} = e^{2\pi i/16}$.
- Good luck!

1. (**6 points**) Execute a DFS on the graph shown below starting at node A and breaking ties alphabetically. Draw the DFS tree/forest. Mark the pre and post values of the nodes with numbering starting from 1.



Node	pre	post
A		
B		
C		
D		
E		
F		
G		
H		
J		

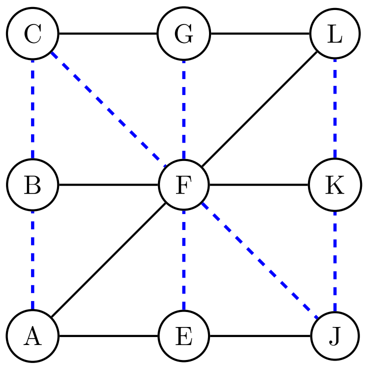
Draw the DFS Tree/Forest in the box below:

2. (**4 points**) In the DFS execution from above, mark the following edges as as **T** for Tree, **F** for Forward, **B** for Back and **C** for Cross.

Edge	Type
$C \to G$	
$J \to E$	
$G \to F$	
$E \to A$	

3. (**4 points**) Draw the DAG of strongly connected components for the above graph.

4. Suppose the dashed edges represent the unique MST in the graph below.



(a) (**4 points**) List all edges necessarily heavier than the edge BC.

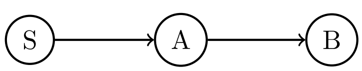
(b) (**4 points**) List all edges that are necessarily lighter than the edge GL.

5. Recall the update operation on distances used in the Bellman-Ford algorithm:

$$\begin{aligned}&\mathit{update}\,(\,\mathit{edge}\,(\,u \to v\,)\,)\\&\qquad \mathit{dist}[v] = \min\,(\,\mathit{dist}[v],\ \mathit{dist}[u] + \ell(u,v)\,)\end{aligned}$$

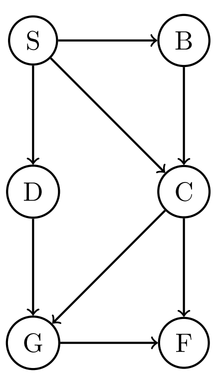
In the following graphs, find the shortest sequence of *update* operations that will ensure that all the distance values from S are correctly computed, irrespective of the weights on the edges. (For simplicity, assume that all edge weights are positive) We have solved the first one as an example.

(a) (Example)

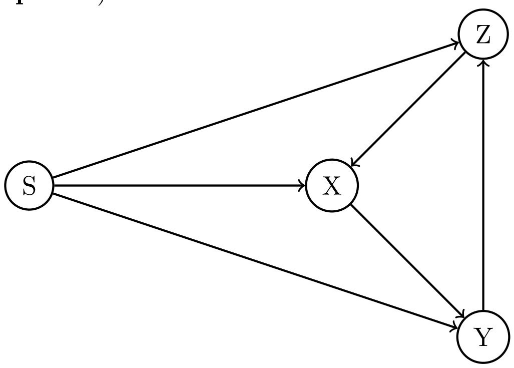


Answer: $S \to A$, $A \to B$

(b) (**6 points**)



(c) (**8 points**)



6. Solve the following recurrence relations:

   (a) (**6 points**)

$$T(n) = 8T(n/2) + O(n^3)$$

$T(n) \in$

Show work:

(b) (**9 points**)

$$T(n) = 2T(\sqrt{n}) + 1$$

$T(n) \in$ $\boxed{}$

Show work:

7. (a) (**5 points**) Suppose we want to multiply the following two polynomials using FFT:

$$A(x) = 3 - x + 4x^2 + 2x^3$$
$$B(x) = -2x^2 + x^3$$

What values would we evaluate the polynomials at to transform them?

(b) (**5 points**) Suppose the FFT of a polynomial $A(x)$ of degree less than 3, at the 4-th roots of unity yields the vector $(1, 1, 1, 1)$. What is $A(x)$?

8. (**5 points**) During the execution of BFS in an unweighted undirected graph G starting from a node S, two vertices A and B are both present in the queue at some point in time. What are the possible values of $dist[S, A] - dist[S, B]$?

9. (**10 points**) Mark the following statements as True (**T**) or False (**F**). Scoring for this part is +2 for correct, 0 for blank, and **-2 for incorrect**.

   (a) Suppose the edge weights are positive and distinct, if an edge e is not the lightest edge across a cut $(S, \overline{S})$, it cannot be part of any MST.

   (b) Suppose the edge weights are positive and distinct, if an edge e is the lightest edge across a cut $(S, \overline{S})$, it is part of some MST.

   (c) Suppose the edge weights are positive and distinct, every edge in an MST is the lightest edge across some cut in the graph.

   (d) Suppose the edge weights are positive and distinct, and e is the lightest edge incident at a vertex v, then e is part of every MST.

   (e) Suppose the edge weights are positive and distinct, and e is the lightest edge incident at a vertex v then e is part of the shortest path tree rooted at v.

10. (**8 points**) Shown below are the pre- post values of vertices A, B, C, D, E in a DFS traversal of a graph G (there are other nodes in G that are not shown).

Node	Pre-value	Post value
A	100	200
B	180	240
C	160	300
D	120	140
E	110	150

(a) One of the vertices has incorrect pre/post values, can you guess which one?

Briefly, justify your answer.

(b) Ignore the vertex with the incorrect pre/post values. List all pairs among the remaining that cannot be edges in the graph G.

(c) Ignore the vertex with the incorrect pre/post values. Suppose G is a DAG, which additional pairs cannot be edges in G (apart from those already listed in the previous question)?

11. (**20 points**) Range Sum
Given integer array $A[1..n]$, design a $O(n \log n)$ algorithm to find the number of ranges $[a, b]$ such that summing the values of that range gives zero, i.e.,

$$\sum_{i=a}^{b} A[i] = 0$$

Describe the main idea behind the algorithm in a few sentences. (No need for proof of correctness or runtime analysis)

12. (**25 points**) The transportation network of Trigonoland consists of n cities (denoted by vertices V) and a network of bus routes represented by edges E_B, train routes represented by edges E_T and airplane routes represented by edges E_A between the cities.

    For every edge $e \in E_B \cup E_T \cup E_A$, let $t(e)$ denote the length of the edge.

    Devise an efficient algorithm to compute the length of the quickest route from a city s to a city t that never uses the same mode of transport in two consecutive edges along the route.

    For example, if the path takes a train from city i to city j, then it can't take a train out of city j.

    (a) **Main Idea**

    (b) **Runtime of algorithm** =

(c) **Proof of Correctness**