Midterm 1

Name:

SID:

Name and SID of student to your left:

Name and SID of student to your right:

Exam Room:
☐ 2040 VLSB ☐ 2050 VLSB ☐ 2060 VLSB ☐ Hearst Field Annex A1
☐ 160 Kroeber ☐ 100 GPB ☐ 10 Evans ☐ 60 Evans ☐ HP Auditorium
☐ 145 Dwinelle ☐ 540AB Cory (6-8 pm) ☐ 540AB Cory (8-10 pm) ☐ Other

Please color the checkbox completely. Do not just tick or cross the box.

Rules and Guidelines

- **The exam is out of 140 points and will last 110 minutes.**
- Answer all questions. Read them carefully first. Not all parts of a problem are weighted equally.
- The first 3 problems are short, true/false, no justification required problems. The next 3 problems are short problems where work is required. The last 3 problems are more involved algorithm design problems.
- 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.
- The problems may **not** necessarily follow the order of increasing difficulty. ***Avoid getting stuck on a problem.***
- Any algorithm covered in lecture can be used as a blackbox. Algorithms from homework need to be accompanied by a proof or justification as specified in the problem.
- 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., $\omega_n = e^{2\pi i/n}$.
- You may assume that comparison of integers or real numbers, and addition, subtraction, multiplication and division of integers or real or complex numbers, require $O(1)$ time.
- Good luck!

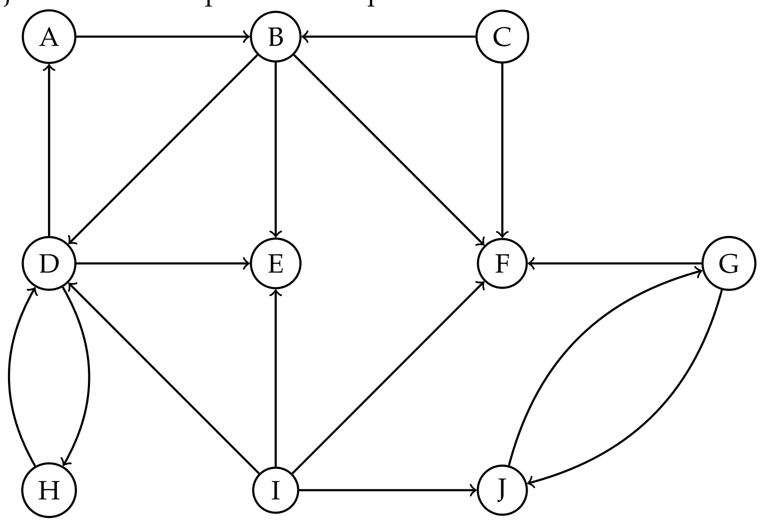
Discussion Section

Which of these do you consider to be your primary discussion section(s)? Feel free to choose multiple, or to select the last option if you do not attend a section. **Please color the checkbox completely. Do not just tick or cross the boxes.**

- ☐ Perla, Monday 9 - 10 am, Dwinelle 243
- ☐ Jenny, Monday 9 - 10 am, Soda 320
- ☐ Sean, Monday 10 - 11 am, Cory 241
- ☐ Yining, Monday 10 - 11 am, Wheeler 222
- ☐ Jerry, Monday 11 am - 12 pm, Etcheverry 3113
- ☐ Jeff, Monday 11 am - 12 pm, Dwinelle 130
- ☐ Peter, Monday 12 - 1 pm, Evans 3
- ☐ David, Monday 12 - 1 pm, Wheeler 108
- ☐ Nate, Monday 12 - 1 pm, Soda 320
- ☐ James, Monday 1 - 2 pm, Dwinelle 182
- ☐ Mudit, Monday 1 - 2 pm, Soda 320
- ☐ Arun, Monday 1 - 2 pm, Barker 110
- ☐ Jierui, Monday 2 - 3 pm, Evans 9
- ☐ Simin, Monday 2 - 3 pm, Evans 70
- ☐ Brandon, Monday 2 - 3 pm, Dwinelle 242
- ☐ Ming, Monday 3 - 4 pm, Cory 289
- ☐ Harley, Monday 3 - 4 pm, Evans 9
- ☐ Aarash, Monday 4 - 5 pm, Dwinelle 79
- ☐ Vinay, Monday 4 - 5 pm, Etcheverry 3119
- ☐ Zheng, Monday 4 - 5 pm, Evans 70
- ☐ Zihao, Monday 5 - 6 pm, Dwinelle 79
- ☐ Max, Monday 5 - 6 pm, Dwinelle 243
- ☐ Matthew, Tuesday 9 - 10 am, Wheeler 108
- ☐ Ajay, Tuesday 9 am - 10 am, Etcheverry 3113
- ☐ Nick, Tuesday 11 am - 12 pm, Etcheverry 3111
- ☐ Sam, Tuesday 12 - 1 pm, Etcheverry 3111
- ☐ Julia, Tuesday 1 - 2 pm, Etcheverry 3119
- ☐ Don’t attend Section.

1 Connectivity in Graphs

No justification is required on this problem.



(a) (**2 points**) For the directed graph above, write down its strongly connected components (SCCs) in any topological order.

Please write the solution in the form (for example) [(ABD), (CE), F, (GH)].

(b) (**3 points**) **True or False:** There exists one edge that would make the graph have exactly two source SCCs and one sink SCC.

○ True ○ False

2 True/False

No justification is needed nor will be examined. DFS means depth first search, and BFS means breadth first search. **(3 points each.)**

(a) n^2 is in $O(n^3)$.

○ True ○ False

(b) Let $T(n) = 4T(n/2) + n$ and $S(n) = 3S(n/2) + n$. Then $T(n) = O(S(n))$.

○ True ○ False

(c) In a DFS on a directed graph $G = (V, E)$, if u was explored before v, and there is a path from u to v, $post(v)$ is greater than $post(u)$.

○ True ○ False

(d) Suppose DFS is called on a DAG G, and the resulting DFS forest has a single tree. Then G has exactly one source vertex.

○ True ○ False

(e) If a root r in a DFS tree of undirected graph G has more than one child, then $G - r$ (the graph resulting from removing r and any incident edges from G) has more than one connected component.

○ True ○ False

(f) Let G be an undirected graph and G^* be the result of removing some edge (u, v) from G. Suppose that G, G^* are connected, and let T be a BFS tree of G rooted at s, and T^* a BFS tree of G^* rooted at s. If for all vertices w, the depth of w in T is the same as the depth of w in T^*, then u and v are at the same depth in T.

○ True ○ False

(g) Let T be any tree which contains all the vertices of a connected undirected graph G. There is a way to break ties in DFS that will output T.

○ True ○ False

(h) Let T be any tree which contains all the vertices of a connected undirected graph G. There is a way to break ties in BFS that will output T.

○ True ○ False

(i) Suppose G is strongly connected, with integer edge weights, and has a shortest paths from some vertex v (i.e. a finite weight shortest path exists to all nodes from v). Then there are shortest paths from every vertex to every other vertex.

◯ True ◯ False

(j) Suppose a directed graph G has integer edge weights and has well-defined shortest path tree (SPT) from some source s (i.e. a finite shortest path exists to all nodes). Bellman-Ford always returns a correct SPT.

◯ True ◯ False

(k) Suppose G is a DAG. The longest path can be found by negating all edge lengths and then running Dijkstra's algorithm from every source node.

◯ True ◯ False

3 Short Answer

No need to justify. Anything outside the box will not be graded unless you **clearly** indicate that your answer is elsewhere (e.g., with an arrow from inside the box). You should try to avoid doing this, though.
(4 points each.)

(a) If each of the integers a and b uses n-bits in their binary representations, what is the **maximum** number of bits that are needed to represent $a \times b$? (An exact expression in terms of n should be given.)

For the following three parts, solve for $T(n)$. Write your answer using Θ notation. Assume $T(c) = \Theta(1)$ for any suitable constant c.

(b) $T(n) = 9T(n/3) + n^2$.

(c) $T(n) = 2T(n/2) + n \log n$

(d) $T(n) = 4T(\sqrt[4]{n}) + \log n$.

(e) What is the largest number of directed edges in an n-vertex DAG?

(f) Given an n-vertex DAG with a single source, what is the maximum number of trees in the DFS search forest among any ordering of the vertices (in DFS)?

(g) Consider an undirected graph with n vertices, and m edges, and maximum degree Δ. Run Dijkstra's algorithm using the array $d[\cdot]$ to maintain distance labels. What is the maximum number of times that $d[v]$ can be decreased for any fixed vertex v?

4 Quick Fixes: Graphs

Your answers here should be one or two sentences. A clear statement of the answer or algorithm is fine.

(a) Consider a depth first search of a **directed** graph G that gives a $pre[\cdot]$ and $post[\cdot]$ numbering and the set of roots of each tree output.

(i) (**4 points**) Consider deleting a single root r and its incident edges from G to form G'. Describe how to modify the pre and post arrays to give valid ones for G' for depth first search where vertices are ordered with respect to the pre-ordering number. (Note that you have no access to G', you just have access to the arrays.)

(ii) (**4 points**) Recall that a forward and tree edge $e = (u, v)$ both have $pre[u] < pre[v] < post[v] < post[u]$. Give a method to distinguish whether an edge $e = (u, v)$ is a forward or tree edge using only the pre and post numberings (i.e., without knowledge of any other edge.)

(b) You are given a directed, strongly connected graph $G = (V, E)$ with **possibly negative** but finite edge weights, $\ell(u, v)$ for directed edge $e = (u, v)$, and an array d indexed by vertices in V.

(i) (**4 points**) Give a linear ($O(|V| + |E|)$) time algorithm to *check* whether the array d corresponds to shortest path distances from a vertex s. (Your algorithm should use $d(\cdot)$, $\ell(u, v)$ for each edge $e = (u, v)$ and an access to $d(v)$ is $O(1)$ time for any vertex v.)

(ii) (**4 points**) Now suppose that $d(\cdot)$ is a valid set of shortest path distances from s. Show how to use Dijkstra's algorithm to compute the shortest distances from some other $s' \in V$. Be sure to describe how to recover the actual distances from your computation.

(Hint: define a new weight for each edge $e = (u, v)$ for the graph G using $d(\cdot)$ and the previous weight $\ell(e)$. Also, notice if $x \geq y$ that $x - y \geq 0$.)

5 Fast Fourier Transform

(a) **(6 points)** Suppose that $FT(a_0, a_2) = (1, 1)$ and $FT(a_1, a_3) = (1, 2)$. What is $FT(a_0, a_1, a_2, a_3)$?
(Recall $FT(a_0, a_1, a_2, a_3)$ is the evaluation of the polynomial $a_0 + a_1 x + a_2 x^2 + a_3 x^3$ on $\{i^0, i, i^2, i^3\}$.)

(b) Let $a_0, \ldots, a_{n-1}$ be some complex numbers.

$$A = (A_0, \ldots, A_{n-1}) = FT(a_0, \ldots, a_{n-1}) \quad \text{and} \quad A' = (A'_0, \ldots, A'_{n-1}) = FT(a'_0, a_1, \ldots, a_{n-1})$$

That is, A' is the FFT of a sequence with exactly one element changed.

(i) **(5 points)** How many values of A and A' differ if $a_0 \neq a'_0$?

(ii) **(5 points)** Consider that $(A_0, \ldots, A_n) = FT(a_0, \ldots, a_{n-1})$, give a $O(n)$ (linear) time algorithm to compute $(A'_0, \ldots, A'_n) = FT(a'_0, a_1, \ldots, a_{n-1})$.

6 A cool set encoding

Given a sequence of integers $S = s_1, \ldots, s_k$, there is a curious way to treat it as a polynomial: $S(x) = \sum_{i=1}^{k} x^{s_i}$. Note that if several elements have the same value v, the x^v term will appear several times and can be collected together. If one wishes to see how many elements of the sequence are equal to v, then one can simply look at the coefficent of x^v in $P(x)$.

Given three sets R, S, T, each of size n and containing integers in the range $[0, n]$, we wish to find the number of solutions to the equation $r + s + t = z$ where $r \in R, s \in S$ and $t \in T$.

(a) (**2 points**) Give a straightforward $O(n^3)$ time algorithm to find the number of solutions to this equation. (Your description should be very brief.)

(b) (**8 points**) Give a more efficient algorithm for this problem. (Here describe your algorithm, argue correctness, and give a runtime bound.)

7 Deterministic Selection

You will design a deterministic algorithm for selecting the kth smallest element of n numbers in some set S. (Assume that the numbers are all distinct. You may describe your method as a modification to an algorithm presented in class or the book.)

(a) (**4 points**) Group the n numbers in S into groups of 7, and take the median of each to form the set M. Now, take the median x of M. Give a lower bound on the number of elements in S that are smaller than x. (You can assume that n is divisible by 7 or not worry about rounding, i.e., you can assume $\lfloor n/7 \rfloor = n/7$)

(b) (**6 points**) Briefly describe a linear time algorithm for computing the kth element of S. If you use a recursive algorithm, please provide a recurrence for the runtime of your algorithm. (Again, for recursive algorithms, don't worry about rounding.)

8 You are not better than me! (Pareto Optimality)

A point (x, y) is dominated by another point (x', y') if $x < x'$ and $y < y'$.

Given a set of points $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, an undominated point $p = (x_i, y_i)$ is one which is not dominated by any other point in S. We wish to find an algorithm for finding all the undominated points. (Assume that for all $i \neq j \in \{1, \ldots, n\}$, $x_i \neq x_j$ and $y_i \neq y_j$, i.e., all x and y values are distinct.)

(a) (**2 points**) Give a straightforward $O(n^2)$ time algorithm for finding the undominated points. (Just the idea here, it should be a sentence.)

(b) (**4 points**) Let m_x be the median value of the x-values of the points and define $S_0 = \{(x, y) \in S : x \geq m_x\}$ and $S_1 = \{(x, y) \in S : x < m_x\}$. That is, S_0 contains all points with x value *at least* the median x-value, and S_1 contains all points with x value *less than* the median x-value.

Can any point in S_0 be dominated by a point in S_1?

(c) (**6 points**) Briefly describe an $O(n \log n)$ time algorithm for finding the set of undominated points. Briefly justify why your algorithm is correct and prove that it runs in the required time.

9 Holiday planning

(10 points)

Alice and Bob are childhood friends, but they now go to different colleges in different cities. Alice lives in city t and Bob in City s. Spring break is coming, and Bob plans to fly to see Alice.

The list of possible flights can be described as a directed graph $G = (V, E)$, and $\{w_e | e \in E\}$ represent the cost of every single flight. As a college student, Bob wants to keep the traveling expense as low as possible. He has c universal coupons, each of which can be used for a 50% discount on any single flight.

Define a shortest path problem whose solution can be used to find the route from s to t with minimal spending. Precisely describe the set of vertices and edges in your graph and which shortest path algorithm you will use, and what you output from that computation. Briefly justify that your solution produces the correct answer, and state its running time in terms of $|V|$, $|E|$ and c.