CS 170 Midterm 1

Write in the following boxes clearly and then double check.

Name :

SID :

Exam Room : ◯ Dwinelle 145 ◯ Hearst Field Annex A1 ◯ VLSB 2050 ◯ Other (Specify):

Name of student to your left :

Name of student to your right :

- **The exam will last 110 minutes.**
- The exam has 10 questions with a total of 100 points. You may be eligible to receive partial credit for your proof even if your algorithm is only partially correct or inefficient.
- Only your writings inside the answer boxes will be graded. **Anything outside the boxes will not be graded.** The last page is provided to you as a blank scratch page.
- Answer all questions. Read them carefully first. Not all parts of a problem are weighted equally.
- Be precise and concise.
- The problems may **not** necessarily follow the order of increasing difficulty.
- The points assigned to each problem are by no means an indication of the problem's difficulty.
- The boxes assigned to each problem are by no means an indication of the problem's difficulty.
- Unless the problem states otherwise, you should assume constant time arithmetic on real numbers. Unless the problem states otherwise, you should assume that graphs are simple.
- If you use any algorithm from lecture and textbook as a blackbox, you can rely on the correctness and time/space complexity of the quoted algorithm. If you modify an algorithm from textbook or lecture, you must explain the modifications precisely and clearly, and if asked for a proof of correctness, give one from scratch or give a modified version of the textbook proof of correctness.
- Assume the subparts of each question are **independent**.
- Please write your SID on the top of each page.
- Good luck!

1 Big O (10 points)

True or false (1 point each).

1. $n^{70} = O(n^{170})$ ◯ True ◯ False
2. $\log(n^{170}) = O(\log(n^{70}))$ ◯ True ◯ False
3. $170^n = O(70^n)$ ◯ True ◯ False
4. $70^n + 170^n = O(240^n)$ ◯ True ◯ False

For the following parts, provide the tightest big O bound that you can.

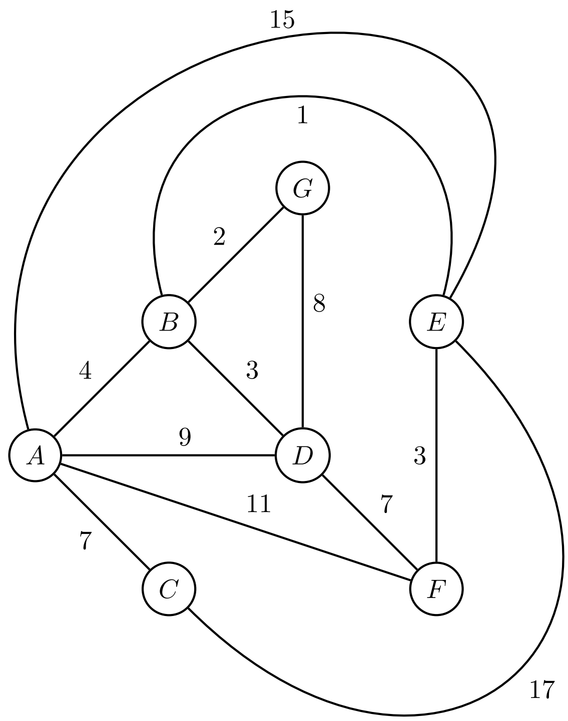
5. (1 point) If $T(n) = 170T\left(\frac{n}{70}\right) + O(n)$, then $T(n) = O(?)$

6. (1 point) If $T(n) = 70T\left(\frac{n}{170}\right) + O(n)$, then $T(n) = O(?)$

7. (4 points) If $T(n) = T(n-1) + 2 \cdot T(n-2)$ (We have $T(1) = 1$ and $T(2) = 2$), then $T(n) = O(?)$
   Note: In case you need to use the quadratic formula, the solutions to the equation $ax^2+bx+c = 0$ are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

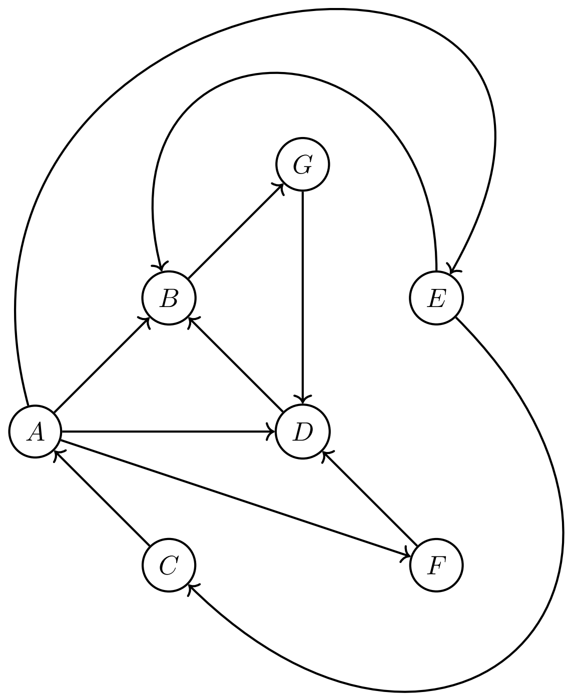
2 Parth's Paths (7 points)



1. (4 points) Run Dijkstra's Algorithm on the above graph for 5 iterations, starting from node C. For each iteration, fill in the below table with the weights of the nodes in the heap as well as the current node that was popped. For vertices that have been popped from the heap already, just fill in the distance to that vertex. The first iteration has been filled in for you already. Break ties in alphabetical order (so node Y before node Z).

Iteration	Curr Node	A	B	C	D	E	F	G
Start	N/A	∞	∞	0	∞	∞	∞	∞
1	C	7	∞	0	∞	17	∞	∞
2								
3								
4								
5								

Exam continues on next page



2. (3 points) Compute all of the strongly connected components (SCC) of the above graph. For ease of grading, please fill in the SCCs in alphabetical order (and the vertices in each SCC also in alphabetical order). For example, if the SCCs were $\{U, V\}$ and $\{T, W\}$, you should write $\{T, W\}$ before $\{U, V\}$. If there are extra boxes left over, you may leave them blank.

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

Exam continues on next page

3 Guess the Number (8 points)

Ajit has a positive integer n, and James has to figure out what it is. James may guess an integer x, and Ajit will say whether $n > x$, $n = x$, or $n < x$. Please help James use the least amount of guesses possible to figure out Ajit's integer.

1. (3 points) If we know $n \leq B$ for some positive integer B, describe an algorithm that uses $O(\log B)$ guesses to guess n.

2. (5 points) If there are no bounds on n (n can be any positive integer), describe an algorithm that uses $O(\log n)$ of guesses to guess n. Please explain its runtime as well.

Exam continues on next page

4 Pairing Param's Pears (8 points)

Param loves pears. He also loves pairing up his pears. Param has $2N$ pears, each located at a distinct integer location on the number line. The locations have already been sorted in increasing order. He wishes to pair up his pears, forming N pairs. However, he would like to minimize the sum of distances between the pears in each pair. Design an efficient algorithm to pair up the pairs while minimizing the sum of distance between each of the pairs, and prove its correctness using an exchange argument. **You do not have to analyze its runtime.**

5 Maximize the Dot Product (10 points)

Richard has an array A of n integers and an array B of m integers with $\log n < m < n$. He wishes to compute the maximum dot product between B and every contiguous subarray of A that has length m. The dot product of 2 arrays p and q (both of length r) is defined as

$$p \cdot q = \sum_{i=1}^{r} p_i q_i$$

where p_i is the i-th element of p (and q_i is the i-th element of q).

1. (1 point) Compute the maximum dot product for $A = [1, 4, 2, 5, 1]$ and $B = [2, -1]$.

2. (9 points) Describe an algorithm to calculate the maximum dot product. Your algorithm must run faster than $O(mn)$ time. Also prove its correctness and analyze its runtime.

6 Igloo Part 1 (7 points)

PNPenguin is in a rectangular igloo, which consists of a $m \times n$ grid of rooms. PNPenguin must travel through all of the rooms. However, each room is separated from each of the 4 adjacent rooms by a door, which requires some amount of energy to open. Once a door is opened, it stays open forever. Rooms on the edge of the grid or the corner of the grid are only connected to the 3 or 2 adjacent rooms, respectively. Let c_{AB} denote the cost of opening a door between rooms A and B (they must be adjacent). PNPenguin starts at the upper left room, located at $(0, 0)$, but doesn't care where he ends up as long as he has visited every room. Design an efficient algorithm to help PNPenguin compute the minimum amount of energy needed to open some set of doors such that PNPenguin can travel through all of the rooms. **You do have to prove its correctness but you do not have to analyze its runtime.**

7 Igloo Part 2 (10 points)

PNPenguin has a rectangular igloo, which consists of a $m \times n$ grid of rooms; each room is separated from each of the 4 adjacent rooms by a door. Rooms on the edge or the corner of the grid are only connected to the 3 or 2 adjacent rooms. Every minute, PNPenguin chooses one of the doors and opens it; now, the two rooms are connected. Every time PNPenguin opens a door, he would like to know how many **new** pairs of rooms are connected (i.e. rooms A and B were not connected before the door was opened, but are connected after the door was opened). Rooms A and B are connected if and only if it is possible to get from room A to room B through a sequence of open doors.

Describe an algorithm that can compute the number of pairs of rooms that are connected. Your algorithm must run in $O(\log(mn))$ time every time a door is opened. Also prove its correctness and analyze its runtime. Your algorithm can store the state of the grid (it doesn't have to start from scratch every time a door is opened), along with other variables if necessary.

8 Shalin's Shuffles (12 points)

Shalin is playing a game where he shuffles a deck of n cards, numbered from 1 to n. Then, for each card c drawn from top of the deck, he gets points equal to the number of cards drawn before c that are of smaller value. Note that by construction of the game he is guaranteed to get 0 points from the first (topmost) card picked. His final score is the sum of scores gained from every individual card.

Given an array A that is the permutation of the deck of cards, with the first element corresponding to the topmost card in the deck, design a divide and conquer algorithm to output the total score from that permutation of cards. Also prove its correctness and analyze its runtime.

Example: if $n = 3$ and $A = [2, 1, 3]$, Shalin's total score is 2. He gets 0 points from drawing 2, 0 points from drawing 1, and 2 points from drawing 3.

9 Maximize the Number (13 points)

Jonathan has an integer n and would like to maximize this integer by performing some sequence of swaps, where two adjacent digits are swapped. For example, if we swap the 1 and the 7 in 710, the resulting number would be 170. However, Jonathan can only perform at most k swaps. Please help Jonathan determine the maximum integer possible after performing at most k swaps.

1. (3 points) If $k = +\infty$, describe an algorithm that runs in $O(\log n)$ time that can determine the largest we can make n. **You do not need to prove its correctness, but you do need to analyze its runtime.**

2. (1 point) If $n = 1263$ and we are allowed 1 swap, what is the largest we can make n?

3. (1 point) If $n = 1263$ and we are allowed 2 swaps, what is the largest we can make n?

4. (8 points) Describe an algorithm that runs in polynomial time (in terms of $\log n$ and k) that can compute the maximum integer possible after performing at most k swaps. **You do not need to prove its correctness nor analyze the runtime.**

10 Wilson's Werewolves (15 points)

n friends are playing a game. k of them are werewolves, while the other $n-k$ of them are villagers. The game is set in a cave with some number of rooms, and rooms may be connected by corridors. At the start of the game, there is a set of broken rooms in the cave that require repairs, and each of the players spawns in a random room. There is no limit on the number of players that can spawn in each room, and it is possible for a player to spawn in a broken room. Everyone then rushes to the closest broken room (determined by number of corridors to get to the room) to either help fix the cave (villager), or pretend to help fix the cave (werewolf). **We want to determine how many villagers are in danger because they end up in the same broken room as a werewolf.**

Assume that the cave can be represented as an unweighted, undirected graph $G = (V, E)$, where the vertices represent rooms and the edges represent corridors connecting rooms. The set of broken rooms is $B \subseteq V$. Further assume that all the players take the shortest path to the closest broken room, and there are no ties. Finally, assume that every room has the capacity to hold any number of players.

Design an efficient algorithm to solve this problem and provide its runtime in terms of any of $n, k, |V|, |E|, |B|$; **proof of correctness is not required.**

Blank scratch page.
This page **will not be graded**.