Midterm 2: November 13, 7-9pm, 2017

A	B	C	D	E	F	G	H	J	K
9am	10am	11am	noon	1pm	1pm	2pm	2pm	3pm	3pm
Rucha	Rucha	Srihita	Shant	Abhishek	Xilin	Shalan	Phillip	Vishal	Phillip
SC 1404	SC 1404	SC 1404	DCL 1320	DCL 1320	DCL 1320	ECE 1002	ECE 1002	ECE 1002	ECE 1002

Name:	
NetID:	
Name on Gradescope:	

- ***Don't panic!***
- Please print your name and NetID **in each page** in the appropriate fields, and circle your discussion section in the boxes above. We will return your exam at the indicated section.
- If you brought anything except your writing implements, your double-sided **handwritten** (in the original) 8½" × 11" cheat sheet, and your university ID, please put it away for the duration of the exam. In particular, please turn off and put away *all* medically unnecessary electronic devices.
  - Submit your cheat sheet together with your exam. An exam without your cheat sheet attached to it will not be graded.
  - If you are NOT using a cheat sheet, please indicate so in large friendly letters on this page.
- Please ask for clarification if any question is unclear.
- **This exam lasts 120 minutes.**
- If you run out of space for an answer, feel free to use the blank pages at the back of this booklet, but please tell us where to look.
- As usual, answering any (sub)problem with "I don't know" (and nothing else) is worth 25% partial credit. Correct, complete, but sub-optimal solutions are *always* worth more than 25%. A blank answer is not the same as "I don't know".
- Total IDK points for the whole exam would not exceed 10.
- **Beware of the Four Deadly Sins.** Give complete solutions, not examples. Declare all your variables. If you don't know the answer admit it and use IDK. Do not write unnecessary text.
- **Style counts.** Please use the backs of the pages or the blank pages at the end for scratch work, so that your actual answers are clear.
- Try to avoid writing in the margins of the pages, as it might not be scanned in.
- Please return ***all*** paper with your answer booklet: your cheat sheet, and all scratch paper. We will **not** return the cheat sheet.
- ***Good luck!***

NETID: NAME:

1 (20 PTS.) Short questions.

1.A. (10 PTS.) Give an asymptotically tight solution to the following recurrence, where $T(n) = O(1)$ for $n < 10$, and otherwise:
$T(n) = \lfloor\sqrt{n}\rfloor T(\lfloor\sqrt{n}\rfloor) + O(n)$.

1.B. (10 PTS.) Given a directed graph G, describe a linear time algorithm that computes a pair of vertices u, v such that there is no path from u to v. If no such pair exists, then the algorithm should output "no such pair".

NETID: NAME:

2 (20 PTS.) (**Seen in lab**) You are given a directed graph G with n vertices and m edges. Every edge $x \to y \in \mathsf{E}(\mathsf{G})$ has a weight $w(x \to y)$ associated with it. Here, exactly *one* edge $u \to v$ has negative weight (all other weights are positive numbers). Describe an algorithm, as fast as possible, that decides if there is a negative cycle in G, and if not, it computes the shortest path between the two given vertices s and t in G. What is the running time of your algorithm? Argue that your algorithm is correct.

NETID: NAME:

3 (20 PTS.) Let $A[1 \,..\, n]$ be an array of n distinct numbers that is not sorted, but it is k-scrambled. Here, being *k-scrambled* means that for any i, we have that $i - k \leq \text{rank}(A[i]) \leq i + k$, where $\text{rank}(A[i])$ is the location of $A[i]$ in the sorted version of A.

You are given the value of k as part of the input, and a number x. Describe an algorithm, as fast as possible, that reports whether or not x is stored in A somewhere.

What is the running time of your algorithm? Explain (shortly) why your algorithm is correct.

(Hint: What can you can say about the relation between $\text{rank}(A[i])$ and $\text{rank}(A[i + 3k])$?)

NETID: NAME:

4 (20 PTS.) There are n people living along Purple street in Shampoo-Bananananana. The n people live in locations specified by a given array $t[1 \,..\, n]$, where $0 < t[1] < \cdots < t[n]$. Here $t[i]$ is the location of the ith person, which is the distance in meters from the start of Purple street,

The district needs to break the street into blocks. A block can have between Δ and 2Δ people living in it, for some prespecified parameter Δ (where $n/3 > \Delta > 1$). A block is a consecutive interval along Purple street. Every person is assigned to a block containing it. A portion of Purple street that has no people living on it, does not necessarily has to be in a block.

The price of a block, covering the ith to jth person, is the total length of the block (in meters), which is $t[j] - t[i]$ (the next block would start at $t[j+1]$). The *total price* of a solution is the total length of the blocks in the solution.

Here is an instance and a suggested solution (of total price 10), with $\Delta = 2$:

$$t = [\underline{1, 3, 4}, \underline{8, 9, 10, 11}, \underline{13, 16, 17}]$$

Describe an algorithm, as fast as possible, that computes the minimum price way of breaking the street into blocks. What is the running time of your algorithm as a function of n and Δ? The algorithm you provide should be iterative (i.e., please do not use dictionary/hashing or recursion).

NETID: NAME:

5 (20 PTS.) You are given a graph G with n vertices and m edges (with $m \geq n$), and a parameter k. In addition, for every vertex $v \in \mathsf{V}(\mathsf{G})$, you are given a flag $r(v)$, which is 1 if v is a *router*, and 0 otherwise. A vertex is ***safe***, if there are at least k distinct routers it can reach (here, think about k as being a small number compared to n).

[Do not use hashing or dictionary data-structure in solving this problem.]

5.A. (10 PTS.) For the case that G is a DAG, describe an algorithm, as fast as possible, that computes all the safe vertices of G. What is the running time of your algorithm? Argue (shortly) that your algorithm is correct.

5.B. (10 PTS.) Describe an algorithm, as fast as possible, for the case that G is a general directed graph. What is the running time of your algorithm?