CS/ECE 374: Algorithms & Models of Computation, Spring 2019
Midterm 2 **Monday, April 8, 7-9pm**

Which exam room to go to based on your discussion section.

ECEB 1002	SC 1404	DCL 1320	ECEB 1013	ECEB 1015
AYA 9am Yipu AYB 10am Xilin AYC 11am Xilin AYD noon Mitch AYE 1pm Ravi AYJ 1pm Shant	AYF 2pm Konstantinos AYG 3pm Robert BYE 3pm Jiaming	AYH 4pm Robert AYK 2pm Shant BYA 9am Zhongyi BYC 1pm Shu	BYB 10am Zhongyi BYF 4pm Jiaming	BYD 2pm Shu

Name:	
NetID:	

⇐ Please PRINT

- ***Don't panic!***
- Please print your name, print your NetID, and circle your discussion section in the boxes above.
- There are five questions – you should answer all of them.
- 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. We will not return or scan the cheat sheets, so photocopy them before the exam if you want a copy.
  - If you are NOT using a cheat sheet, please indicate so in large friendly letters on this page.
- Please read all the questions before starting to answer them. Please ask for clarification if any question is unclear.
- **This exam lasts 120 minutes.** The clock started when you got the exam.
- 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 **slightly** sub-optimal solutions are *always* worth more than 25%. Solutions that are exponentially (or significantly) slower than the expected solution would get no points at all. A blank answer is not the same as "I don't know".
- Total IDK points for the whole exam would not exceed 10.
- Give complete solutions, not examples. Declare all your variables. If you don't know the answer admit it and use IDK. Write short concise answers.
- **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.
- Please return ***all*** paper with your answer booklet: your question sheet, your cheat sheet, and all scratch paper.
- ***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:
$T(n) = T(2n/3) + T(n/2) + O(n^2)$.

1.B. (10 PTS.) Given a directed graph G, describe a linear time algorithm that decides if there are three distinct vertices x, y, z, such that (i) there is a path from x to y in G, (ii) there is a path from y to z in G, and (iii) there is a path from z to x in G.

2 (20 PTS.) Given a directed graph $\mathsf{G} = (V, E)$ with positive edge lengths. Let $\ell(u, v)$ be the length of edge $(u, v) \in E$, and let $d(u, v)$ be the length of the shortest path from u to v in G. Given two nodes s and t, there might be many different paths that realize the shortest path between s and t, and let Π be the set of all such paths. A vertex is ***useful*** if it lies on any path of Π. Describe how to compute, as fast as possible, all the useful vertices in G (given s and t). What is the running time of your algorithm.

3 (20 PTS.) Suppose you are given a sorted array of n distinct numbers that has been rotated right by k steps, for some *unknown* integer k between 1 and $n - 1$. That is, you are given an array $A[1...n]$ such that some prefix $A[1...k]$ is sorted in increasing order, the corresponding suffix $A[k + 1...n]$ is also sorted in increasing order, and $A[n] < A[1]$.

For example, the below array with $n = 10$ has been rotated by $k = 7$.

35	65	108	197	303	499	833	3	4	19

Given a number x, describe an algorithm, as fast as possible, that decides if x appears somewhere in the A. What is the running time of your algorithm? Argue that your algorithm is correct.

4 (20 PTS.) We are given a sequence of n numbers $A[1], \ldots, A[n]$, and integers g and ℓ with $\ell \geq n/g$. We want to choose a subsequence $A[i_1], \ldots, A[i_\ell]$ of length ℓ, such that $i_1 = 1$, $i_\ell = n$, and $1 \leq i_{j+1} - i_j \leq g$ for all $j = 1, \ldots, \ell - 1$, while minimizing the sum $A[i_1] + \cdots + A[i_\ell]$.

Example: for the input sequence $0, 4, 3, 1, 11, 8, 5, 2$ and $g = 3$ and $\ell = 5$, we could pick $0 + 1 + 8 + 5 + 2 = 15$, but the optimal solution has sum $0 + 3 + 1 + 5 + 2 = 11$.

Describe an algorithm, as fast as possible, to compute the optimal sum, by using dynamic programming. (You do not need to output the optimal subsequence.) Give a clear English description of the function you are trying to evaluate, and how to call your function to get the final answer, then provide a recursive formula for evaluating the function (including base cases). If a correct evaluation order is specified clearly, pseudocode is not required. Analyze the running time as a function of n, g, and ℓ.

5 (20 PTS.) You are given a directed graph G with n vertices and m edges ($m \geq n$), where each edge e has an integer weight $w(e)$ (which could be positive or negative) and each vertex is marked "red" or "blue". You are also given a (small) positive integer b.

Describe an algorithm, as fast as possible, to find a walk with the smallest total weight, such that the start vertex is red, the end vertex is red, and the number of blue vertices is divisible by b (with no restrictions on the number of red vertices). Your solution should involve constructing a new graph and applying a known algorithm on this graph. Analyze the running time as a function of n, m, and b.