COMPSCI330 Design and Analysis of Algorithms
Midterm Exam

Guidelines

- **Describing Algorithms** If you are asked to provide an algorithm, you should clearly define each step of the procedure, and then analyze its overall running time. There is no need to write pseudo-code; an unambiguous description of your algorithm in plain text will suffice. If the running time of your algorithm is worse than the suggested running time, you might receive partial credits.
- **Timing** Exam starts at 3:05 pm and ends at 4:20 pm.

Name: ____________________________

Duke ID: ____________________________

Problem 1 (Recursions)**.** Please solve the following recursions (write the answer in asymptotic notations $T(n) = \Theta(f(n))$).

If you decide to use the recursion tree method, please draw the tree this time because it's easy to do on a piece of paper. You need to bound the amount of work in each layer, and take the sum over all layers. You do not need to write the induction proof if you are using the recursion tree method.

(a) (10 points)

$$T(n) = 2T(\frac{n}{2}) + 8T(\frac{n}{4}) + n^2.$$

(Base case: $T(1) = 0$)

(b) (15 points)

$$\begin{aligned} T(n) &= F(\sqrt{n}) + n; \\ F(k) &= F(k-1) + T(k). \end{aligned}$$

(Base case: $T(2) = 0, F(0) = 0$)

(Hint: $\sum_{i=1}^{\sqrt{n}} i \approx n/2$, you can assume they are equal for this problem.)

Problem 2 (Flat Point)**.** You are given an array $a[]$ of n numbers ($n \geq 2$). It satisfies the following property (called convexity): for any i, j

$$a[i-j] + a[i+j] \geq 2a[i].$$

One such array is $a[] = \{10, 3, 1, 0, 2, 8\}$. Your goal is to find the flattest point in the array: an index i such that $|a[i] - a[i+1]|$ is as small as possible. In the case above your answer should be 3 as $|a[3] - a[4]| = 1$ is the smallest (if there are multiple solutions you only need to output any one of them). Your algorithm should run in time $O(\log n)$.

(a) (8 points) Design an algorithm to find the flattest point.
(Hint: What can you say about $a[i] - a[i-1]$ and $a[i+1] - a[i]$?)

(b) (12 points) Prove the correctness of your algorithm.

(This page is left empty intentionally.)

Problem 3 (Cellphone Base Stations Revisited)**.** Recall the 3rd problem in problem set 3a: Let's consider a long, quiet country road with houses scattered very sparsely along it. (We can picture the road as a long line segment, with an eastern endpoint and a western endpoint.) More concretely, assume the road has length m, with n houses at coordinates $x_1, x_2, ..., x_n \in [0, m]$. Further let's suppose that despite the bucolic setting, the residents of all these houses are avid cellphone users. You want to place cellphone base stations at certain points along the road so that every house gets served.

What's different now is that the cellphone company has updated their technology: they can now build stations that serve an arbitrary segment of the road. A base station that serves segment $[s, t]$ costs $a + b(t - s)^2$, where $a, b > 0$ are constants given as inputs.

Your goal is to design an algorithm that build enough base stations to cover every house at the minimum cost.

(a) (15 points) Since naïve greedy fails, we will use dynamic programming. Define the state (sub-problems), write the transition function and specify the base cases.

(b) (10 points) Design an algorithm for finding the optimal cost to build base stations. (Your algorithm only needs to output the optimal cost. You need to analyze the running time and you will not receive full credit if it is slower than $O(n^2)$. Note that m can be much larger than n so $O(mn)$ counts as slower.)

(c) (5 points) Prove the correctness of the algorithm you designed in (b).

(This page is left empty intentionally.)

Problem 4 (Shared-Bus)**.** Famous ride-sharing company Ubyft has created a new "shared-bus" service. The bus serves a path of m miles, indexed by $[0, m]$. The main differences between this service and traditional bus services are:
1. All the passengers board the bus at point 0.
2. Passenger i specifies a range $[s_i, t_i] (0 \leq s_i \leq t_i \leq m)$. The passenger is satisfied as long as the bus stops at some location between s_i and t_i (including the boundary).

There are no fixed bus stops. Instead, the stops will be decided after gathering the user information. As an engineer at Ubyft, you need to solve the following problem: given the number of passengers n, and each passenger's preference $[s_i, t_i]$, design an algorithm that finds the smallest number of stops in order to satisfy all customers.

(a) (10 points) Design the algorithm and analyze its running time. Your algorithm should run in $O(n \log n)$ time.

(b) (15 points) Prove the correctness of the algorithm you designed in (a).

(This page is left empty intentionally.)