COMS 3110: Homework 5
Due: August 5^{th}, 11:59pm
Total Points: 75

Submission format. Homework solutions will have to be typed. You can use word, LaTeX, or any other type-setting tool to type your solution. Your submission file should be in pdf format. Do **NOT** submit a photocopy of handwritten homework except for diagrams that can be hand-drawn and scanned. We reserve the right **NOT** to grade homework that does not follow the formatting requirements. Name your submission file: `<Your-net-id>-3110-hw5.pdf`. For instance, if your netid is `asterix`, then your submission file will be named `asterix-3110-hw5.pdf`. Each student must hand in their own assignment. If you discussed the homework or solutions with others, a list of collaborators must be included with each submission. Each of the collaborators has to write the solutions in their own words (copies are not allowed).

General Requirements

- When proofs are required, do your best to make them both clear and rigorous. Even when proofs are not required, you should justify your answers and explain your work.
- When asked to present a construction, you should show the correctness of the construction.

Some Useful (in)equalities

- $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$
- $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$
- $2^{\log_2 n} = n$, $a^{\log_b n} = n^{\log_b a}$, $n^{n/2} \leq n! \leq n^n$, $\log x^a = a \log x$
- $\log(a \times b) = \log a + \log b$, $\log(a/b) = \log a - \log b$
- $a + ar + ar^2 + ... + ar^{n-1} = \frac{a(r^n-1)}{r-1}$
- $1 + \frac{1}{2} + \frac{1}{2^2} + ... + \frac{1}{2^n} = 2(1 - \frac{1}{2^{n+1}})$
- $1 + 2 + 4 + ... + 2^n = 2^{n+1} - 1$

---

1. **(20 points)** Design an efficient algorithm to solve the following problem. The algorithm that takes input the undirected graph $G = (V, E)$, weight function $w : E \to \mathbb{Z}$, a minimum spanning tree $T \subseteq E$ with respect to w, an edge $\eta \in T$, and a number $\alpha \in \mathbb{N}$. It should compute and return a minimum spanning tree T' of G with respect to the weight function $w' : E \to \mathbb{Z}$ where

$$w'(e) = \begin{cases} w(e) & e \neq \eta \\ w(e) + \alpha & e = \eta \end{cases}$$

Analyze the runtime of your algorithm.

2. **(25 points)** Solve the following recurrence equations using the recurrence tree method. You must show the table as seen in lecture, but you do not need to draw a picture of the tree.

   (a)
$$T(n) = \begin{cases} T(n/5) + cn & (n > 1) \\ 1 & (n = 1) \end{cases}$$

   (b)
$$T(n) = \begin{cases} 16T(n/2) + n^3 & (n > 1) \\ 1 & (n = 1) \end{cases}$$

3. **(30 points)** You're playing a game, and your character regularly encounters the following puzzle:

   The character is placed in the northwest corner of a $M \times N$ grid, room $0, 0$. All of the rooms in the southernmost row, $M - 1$, have ladders allowing him to escape. Each room in the grid has a one-way door to each room directly south, east, and southeast of it. To pass through one of these doors, the character must pay a fee. These fees are 3 credits for a south door, 2 for a southeast door, and 1 for an east door. Each room in the grid also has a chest that can be opened without cost, awarding the character between 0 and 10 credits.

   (a) Give a recurrence relation that describes the optimal solution to the puzzle. The optimal solution is one that gets the character to an exit room with the greatest possible amount of credits.

   (b) Sometimes, your character is granted foreknowledge of the credits hidden in each room's chest. Design an efficient dynamic programming algorithm that takes as input C, the $M \times N$ table of credits, and returns the optimal escape route.

   (c) Analyze the worst-case runtime of your algorithm.