# CS41 Homework 8

This homework is a **2 week, 20 point** assignment due at 11:59PM on Wednesday, November 16. Write your solution using LaTeX. Submit this homework in a file named `hw8.tex` using **github**.

This is a partnered homework. You should primarily be discussing problems with your homework partner. It's ok to discuss approaches at a high level with others. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab teammate *while in lab*. In this case, note (in your **homework submission poll**) who you've worked with and what parts were solved during lab.

1. **Implementing RNA Substructure Dynamic Program**

   For this problem, your task is to write a dynamic program that takes as input an RNA string and outputs the size of the largest matching, according to the rules of the RNA Substructure problem described in class.

   - You may write your program in C++ or in Python.
   - If you write in C++, we encourage you to use the limited starting code we've provided.
   - Your solution must be a dynamic program.
   - Your solution must be efficient—graders will test your program against large RNA substrings, and there will be nontrivial penalties for programs that take too long.
   - The file format consists of a single string of letters from $\{A, C, G, U\}$. Your program should read in the file, compute the largest RNA Subsequence matching, and output the result.

2. **Cassie's Convenience Stores.** Carrie plans to open a chain of convenience stores along Baltimore Pike. Using market research, Carrie identified a series of $n$ locations where she can open stores. For each location, Cassie calculated (again using market research) how much annual profit she is likely to gain by placing a store at this location. She can build as many convenience stores as she wants, as long as they are not too close (otherwise, they will compete with each other for business and lose money).

   In this problem, you will design an algorithm that helps Cassie determine how much annual profit she can make. The input to this problem is an integer $K$, and two arrays $L[1 \cdots n]$ and $P[1 \cdots n]$. Assume that $0 \leq L[i] \leq N$ for each $i$[1], and that $L$ is sorted in increasing order. The goal of this problem is to output the maximum possilbe profit by placing convencience stores at locations from $L[1 \cdots n]$ such that the distance between any two locations is at least $K$.

   (a) If Cassie decides to build a convenience store at location $L[k]$, what is the closest location to the east or west that she can build, given her list of locations? Write an algorithm $West[k]$ that returns the index of the closest location $k'$ to $k$ such that $L[k'] < L[k]$. Write a similar algorithm for $East[k]$.

[1] $L[i]$ represents the the location of store $i$, where 0 is the westernmost terminus of Baltimore Pike, and $N$ is the easternmost terminus

(b) Design a dynamic program that computes the maximum annual profit Cassie can earn by placing her convenience stores.

(c) Modify your dynamic program so it returns the set of locations that maximize Cassie's profit.

3. **Moving on a Checkerboard.** Suppose you are given an $n \times n$ checkerboard and a single checker. You must move the checker from the bottom edge of the board to the top edge of the board according to the following rule: at each step, you may move the checker to one of the following three squares:

   ($a$) the square immediately above,

   ($b$) the square that is one up and one to the left (but only if the checker is not already in the leftmost column)

   ($c$) the square that is one up and one to the right (but only if the checker is not already in the rightmost column)

   Each time you move from square $x$ to square $y$, you receive $P(x, y)$ dollars. You are given the values $P(x, y)$ for all pairs $(x, y)$ for which a move from $x$ to $y$ is legal. $P(x, y)$ may be negative.

   Give a polynomial-time algorithm that computes the set of moves that will move the checker from somewhere along the bottom edge to somewhere along the top edge while gathering as many dollars as possible. Your algorithm is free to pick any square along the bottom edge as a starting point, and any square along the top edge as an ending point, in order to maximize the number of dollars gathered along the way.

   What is the runtime of your algorithm?

4. **Flow variant.** In the standard flow problem, we get an input $G = (V, E)$ a directed graph and edge capacities $c_e \geq 0$ limiting how much flow can pass along an edge. Consider the following two variants of the maximum flow problem.

   Your algorithms should be a reduction to the Ford-Fulkerson algorithm. Be careful about the output formatting!

   (a) It might be that each junction where water pipes meet is limited in how much water it can handle (no matter how much the pipes can carry). In this case, we want to add *vertex* capacities to our problem. The input is a directed $G$ (with source $s$ and sink $t \in V$), edge capacities $c_e \geq 0$, and vertex capacities $c_v \geq 0$ describing the upper limit of flow which can pass through that vertex. Give a polynomial-time algorithm to find the maximum $s \rightsquigarrow t$ flow in a network with both edge and vertex capacities.

   (b) It might be that there are multiple sources and multiple sinks in our flow network. In this case, the input is a directed $G$, a list of sources $\{s_1, \ldots, s_x\} \subset V$, a list of sinks $\{t_1, \ldots, t_y\} \subset V$, and edge capacities $c_e \geq 0$.
   Give a polynomial-time algorithm to find the maximum flow in a network with multiple sources and multiple sinks.

5. **Hospitals coping with natural disaster.** (K&T 7.9)

   The same hospitals from earlier in the semester have now hired all the doctors they need. There is a widespread natural disaster (like the fires in California!), and a lot of people across an entire region need to be rushed to emergency medical care. Each person should be brought to a hospital no more than 50 miles away from their current location. Additionally, we want to make sure that no single hospital is overloaded, so we want to spread the patients across the available hospitals. There are $n$ people who need medical care and $h$ hospitals; we want to find a way to coordinate emergency medical evacuations so that each hospital ends up with at most $\lceil n/h \rceil$ patients in emergency care. (Also, obviously: every patient should end up at a hospital!)

   Give a polynomial-time algorithm that takes the given information about patients' locations and hospitals and determines whether this is possible. If it is possible, your algorithm should also output an assignment of patients to hospitals ensuring that every patient gets to a nearby hospital and that no hospital is overloaded.

   Prove that your algorithm is correct.

6. **(extra challenge) Cassie's Convenience Stores (v2.0)** It is natural to assume the profit of a convenience store changes depending on how close to other convenience stores it is. Suppose instead of an array of profits $P[1 \cdots n]$ and a minimum distance between stores $K$, Cassie does more market research and gets information on $P[k, d]$, the annual profit of location $k$ assuming that the closest other store is at least $d$ km away.

   Design an algorithm that computes Cassie's maximum profit.