Math 182 Homework #6

Nathan Solomon

March 6, 2025

Problem 0.1.

Homework 6

Math 182, Winter 2025

When designing an algorithm, you must include the following:

1. A short (1-2 sentence) sketch of the main idea of the algorithm.
2. The algorithm itself, written in pseudo-code.
3. The runtime of the algorithm.
4. A proof that the algorithm is correct.
5. A proof that the runtime is correct.

If you wish to sort a list, assume that you can do so in $O(n \log n)$ time; you do not need to write out the details of the sorting algorithm. If you wish to use an algorithm we have discussed in class (eg BFS/DFS, Interval Scheduling, Dijkstra's Algorithm, Ford-Fulkerson, etc) you may do so without elaboration; if you need to modify the algorithm, you should provide sufficient details for the modification in your pseudocode.

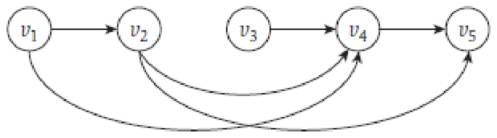
Question 1:

Let $G = (V, E)$ be a directed graph with vertices $v_1, \ldots, v_n$. We say G is an *ordered graph* if it has the following properties:

1. Each edge goes from a node with a lower index to a node with a higher index. That is, every edge (v_i, v_j) must have $i < j$.
2. Every node except v_n has at least one edge leaving it.

Design an efficient algorithm to compute the length (number of edges) of the longest path from v_1 to v_n.

Example: In the graph below, the length of the longest path is 3: v_1, v_2, v_4, v_5.



Question 2:

Suppose you are designing a word-processing program. You would like lines of text displayed by the program to be as even as possible, while not exceeding the margins of the paper. Your task is to design an efficient algorithm to decide how many words to print on each line.

Let's make this more precise. Suppose you are given a list of n words, with lengths $c_1, c_2, \ldots, c_n$ (i.e. the i^{th} word has c_i characters). For simplicity, we will assume that your program uses a monospace font, so all letters are the same width. You are also given a maximum line length L. You need to decide where to insert line breaks so that no line exceeds the maximum line length. That is, you need to find integers $0 = l_1 < l_2 < \ldots < l_k = n$ such that the following condition holds: for all $i \leq k$,

$$\sum_{l_i < j < l_{i+1}} (c_j + 1) + c_{l_{i+1}} \leq L.$$

The idea is that the sum above corresponds to the length of a line which contains words $l_i + 1$ through l_{i+1} and we need to put a space after each word in a line except for the last one. For each $i \leq k$, define the *slack* of line i, which we will denote s_i, to be the difference between L and the sum in the expression above—this corresponds to the number of empty spaces at the end of line i if we use line breaks $l_1, \ldots, l_k$. You need to design an efficient algorithm which finds a list of valid line breaks such that

$$\sum_{i \leq k} s_i^2$$

is as small as possible.

Question 3:

The residents of the underground city of Zion defend themselves through a combination of kung fu, heavy artillery, and efficient algorithms. Recently they have begun developing automated defenses to help thin out swarms of attacking robots before they can reach the city.

The robot swarm arrives at the outer layer of the city's defenses over the course of n seconds. In the i-th second, x_i robots arrive. Based on remote sensing data, the sequence $x_1, \ldots, x_n$ is known in advance.

Zion has installed an device to produce electromagnetic pulses (EMPs), which can be activated to destroy some of the robots when they reach the outer defenses. The EMP's power depends on how long it has been allowed to charge. More specifically, there is a function f so that if j seconds have passed since the EMP was last used, it is capable of destroying up to $f(j)$ robots.

That is, if the EMP is used at second k, and it has been j seconds since its last activation, it will destroy $\min(x_k, f(j))$ robots. We assume that the EMP starts off completely drained, so if its first use is second j, it can destroy at most $f(j)$ robots.

Example: Suppose $n = 4$, and the values of x_i and $f(i)$ are given by the following table:

i	1	2	3	4
x_i	10	100	100	10
$f(i)$	10	20	40	80

The best solution will be to activate the EMP at seconds 3 and 4. This will destroy $40 + 10 = 50$ robots. If we wait until second 4 to fire the EMP for the first time, we will only destroy 1, since all the earlier waves will have already bypassed our defenses. If we fire it earlier, the best we can do is 40 total destroyed robots.

Given the sequence $x_1, \ldots, x_n$ and the function f, design an efficient algorithm that returns the maximum number of robots that can be destroyed by a sequence of EMP activations.

Question 4:

Suppose G is a network which has weights on vertices as well as edges. Design an efficient algorithm to solve the maximum network flow problem in this new setting.

More formally: you are given a network $G = V, E, s, t, w$, as well as a function $c : V \to \mathbb{N}$ assigning a natural number to each vertex. A flow f on G is defined as usual except that it must also satisfy an extra constraint: for any vertex $v \in V$, we must have

$$\sum_{\{u:(u,v)\in E\}} f(u, v) \leq c(v).$$

The size of f is defined as usual; you need to design an algorithm to find a flow of maximum size subject to this extra constraint.

Hint: Modify G to turn this into a normal network flow problem. You may use the standard Ford-Fulkerson algorithm for maximizing network flow without needing to provide details.