CS 341, Fall 2025 T. Brown, É. Schost

ASSIGNMENT 3

DUE: Wed Oct 29, 11:59 PM. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read `http://www.student.cs.uwaterloo.ca/~cs341` for general instructions and policies. **Note:** All logarithms default to base 2 (i.e., $\log x$ is defined as $\log_2 x$). **Note:** "Giving" an algorithm means doing the four parts (i)–(iv) as described on the course web page.

1. [20 marks] **Dynamic Programming + programming question.** In this question, we consider special directed graphs $G = (V, E)$, where for every vertices $a \neq b$ in V, the edge (a, b) is in E, but we have no edge of the form (a, a). We will call a *simple cycle* a walk $a \to b \to c \to \cdots \to v \to a$, where no vertex is repeated except for a.

   We assume that all edges (a, b) have a weight $w[a][b]$ (nonnegative integer), and we call weight of a walk the sum of the weights of its edges. The goal is to find a simple cycle that visits all vertices (so there are n edges on it, $n = |V|$) and has maximal weight. This problem is NP-complete, so it is likely that there is *no polytime algorithm*. Throughout, we use the unit cost model.

   (a) [2 marks] Give a brute force algorithm with runtime $O(n^k n!)$, k constant (you have to find k, of course).

   (b) [7 marks] For a, b two vertices in V (including the case $a = b$), and for S a subset of $V - \{a, b\}$, consider all walks starting at a, going through all vertices of S exactly once (using only vertices from S), and ending at b. Let $W_{a,b,S}$ be this set, and let $w_{a,b,S}$ be the maximum weight of the walks in $W_{a,b,S}$. **Give a dynamic programming algorithm** that computes all $w_{a,b,S}$ (you may want to store extra information as well) **and analyze its runtime**.

   Don't forget to tell us what data structure you use for S, and how you access entries in your DP table.

   You will get full credit if the runtime is $O(n^\ell 2^n)$, for some constant ℓ.

   (c) [1 mark] Using the previous question, explain how to find a simple cycle of maximal weight. Give the runtime for the whole procedure.

   (d) [10 marks] Implement in C++ the algorithms described so far (and submit via Marmoset). The format for input files is:

   - n on the first line; we assume that the vertices are $0, \ldots, n-1$
   - n rows, the ith of which (for $i = 0, \ldots, n-1$) gives you $w[i][0], \ldots, w[i][n-1]$, separated by spaces ($w[i][i]$ is undefined, in our input it will be set to zero)
   - $t \in \{0, 1\}$

   If $t = 0$, run the brute-force algorithm. If $t = 1$, run the DP algorithm. Then, the format for the output is

   - 1st line: the maximal weight of a simple cycle

- 2nd line: a simple cycle that realizes the maximum (do not worry about which vertex you should choose to start, or how to break ties if any; any cycle that gives the maximum is OK). Print the vertices separated by spaces, with the first one repeated the end, and finish your line with a return.

Sample input:

```
4
0 10 33 4
1 0 3 10
0 8 0 8
11 1 1 0
1
```

with possible output (two lines)

```
62
0 2 1 3 0
```

You can assume $n < 32$.

2. [10 marks] **Graphs.** Given a directed graph $G = (V, E)$, with $|V| = n$ and $|E| = m$, and two vertices a, b in G, give an algorithm that returns the number of **shortest** walks from a to b in G. Give its runtime in terms of n and m.