CS 580 Algorithm Design, Analysis, And Implementation
Fall 2021 Vassilis Zikas HW 3

Due Friday Oct. 15 at 11:59 p.m.

Important Note: *You are **only allowed one late day** for this assignment i.e. last date to submit the assignment (with late penalty) is 11:59 pm Monday, October 18. In particular, the deadline to submit without any penalty is still 11:59 pm Friday, October 15, and any submission made after 11:59 pm Friday, October 15 and before 11:59 pm Monday, October 18 will receive a 10% reduction in the grade.* **No submissions will be accepted/graded after 11:59 pm Monday, October 18!**

1. (30 points, 10 + 20)

   (a) You are given an undirected graph G with c connected components and positive edge weights. Consider the set S of subgraphs [1] of G such that any element H of S has $n-c$ edges and the property that if any two vertices are connected in G then they are also connected in H. Find and analyze an efficient algorithm that finds an element (subgraph of G) of S with maximum sum of weights of edges.

   (b) Janet was given a square matrix of size $n \times n$ with entries 0's and 1's, the goal was to come up with an efficient divide and conquer algorithm to find the length of the longest sequence of 1's in the matrix. The sequence can be horizontal or vertical, in other words, it can be contiguous sequence of 1's in a row or in a column. Note that a combination of both horizontal and vertical is NOT valid, for example, L shaped figures are not to be considered. Help Janet design and analyze an efficient divide and conquer algorithm, perhaps as fast as $O(n^2)$, that finds the longest sequence of 1's.

2. (40 points, 5 + 5 + 15 + 15 respectively) Mario is playing a game that is modeled as a directed graph, $G:(V,E)$. Each node $u \in V$ is associated with a positive value L_u which denotes the number of lives Mario gains when she reaches that vertex. Each edge $e \in E$ is associated with a negative cost C_e which denotes the lives required to traverse the edge. Mario can traverse an edge only if she has enough lives to do it. She starts the game at level $V_s \in V$ and needs to reach $V_t \in V$ with maximum possible lives.

   For example, if Mario is at node $u \in V$ with 50 lives (this includes lives collected at node u), and $C_{u,v} = -10$, then Mario will have 40 lives on reaching v (this excludes the lives she collects from v) if she chooses to traverse edge (u,v). If $C_{u,v} = -55$, then she can not traverse the edge because she only has 50 lives. Note that she gets the lives at node v only after reaching it.

   We are also guaranteed that there will be no directed cycles in the graph.

   (a) First, consider the following case. $L_{V_s} = E_0$ i.e. Mario collects E_0 lives at the first node itself, however, for all other vertices $u \in V \setminus V_s$, $L_u = 0$ and Mario only has to spend 1 life to go between any two nodes

[1] Recall that a graph H is called a subgraph of graph G if $V(G) = V(H)$ and $E(H) \subseteq E(G)$. For instance, consider the graph G on 4 vertices, $V(G) = \{a,b,c,d\}$ and edge set $E(G) = \{(a,b),(b,c),(c,d),(d,a)\}$. Then the graph H with the vertex set $V(H) = \{a,b,c,d\}$ and edge set $E(H) = \{(a,b),(c,d)\}$ is a subgraph of G. However, the graph H' with the vertex set $V(H') = \{a,b,c\}$ is not a subgraph of G because the vertex set of H' and G are different.

(i.e. $C_{u,v} = -1$ for all u and v with an edge from u to v). Briefly describe an efficient algorithm that returns the optimal (maximal) number of lives that Mario will have when reaching node V_t. Give the time and space complexity of your algorithm.

(b) Now, Mario spends different amounts of lives to go from one node to another. $L_{V_s} = E_0$ i.e. Mario collects E_0 lives at the first node itself, however, for all other vertices $u \in V \setminus V_s$, $L_u = 0$. Briefly describe an efficient algorithm that returns the optimal (maximal) number of lives that Mario will have when reaching node V_t. Give the time and space complexity of your algorithm.

(c) Now, lives at each node could also be positive. Describe an efficient algorithm that returns the optimal (maximal) number of lives that Mario will have when reaching node V_t. Give the time and space complexity of your algorithm.

Hint: Maybe there is a way to define the problem in terms of subproblems.

(d) This time, the game is exactly the same as the previous version - negative edge costs and non-negative vertex lives - except Mario begins the game with a single skip superpower. She can use the superpower at any time to jump to any node she wants (including nodes already traversed, except for the final node V_t) and recollect the lives at any nodes she reaches after using the superpower. Modify your previous algorithm to account for this case. Give the time and space complexity of your algorithm.

3. (30 points) Consider the following "Balls and Bins" problem. You are given n pairs of balls (i.e. $2n$ balls in total), denoted by $\{A_{11}, A_{12}, A_{21}, A_{22}, \dots A_{n1}, A_{n2}\}$ where $\{A_{i1}, A_{i2}\}$ represent a pair, and m buckets, denoted by $\{B_1, B_2, \dots B_m\}$. We want to allocate balls to buckets. The goal is to maximize the total number of balls assigned to some bucket (some balls will be left un-assigned) such that the following conditions are satisfied.

   - For each ball, say A_{ij} where $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2\}$, there is a list of buckets associated, denoted by P_{ij}. A_{ij} can be assigned to a bucket B_k only when $B_k \in P_{ij}$.
   - Bucket B_i can take at most b_i balls.
   - For balls that form a pair i.e. for any i, the balls $\{A_{i1}, A_{i2}\}$ cannot be assigned to the same bucket.
   - Each ball goes into at most one bucket.

   Design and analyze an efficient algorithm to find the maximum number of balls that can be assigned.

4. **2 points** Have you assigned pages on Gradescope?