Due: 25th of September 2017 at 11:59pm

COMP 2907 – Assignment 3

All submitted work must be done individually without consulting someone else's solutions in accordance with the University's Academic Dishonesty and Plagiarism policies.

IMPORTANT! Questions 1(a–c) and 2(a–c) should be submitted via Blackboard as PDF (no handwriting!). The implementation required for Question 2d and should be done in Ed, and submitted via Ed.

Questions

1. [**25 points**] Consider a sequence of houses H arranged in a straight line, where all houses have an associated positive value $w : H \to \mathbb{R}^+$ (Fig 1).

   Design an algorithm that chooses the subset of houses with maximum combined value, given the constraint that no two chosen houses can be adjacent. That is, you can't choose both a house and its neighbour.

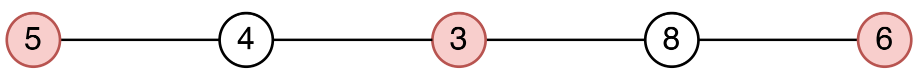

Figure 1: Example input where $n = 5$. The optimum choice of houses is the first, third and fifth house. The value of this selection is $5 + 3 + 6 = 14$.

   (a) Design a dynamic programming algorithm to determine the value of the optimal choice of houses. First derive and justify a recurrence relation and your base cases, then turn it into a bottom-up solution. A more efficient algorithm gives more points. [15 points]

   (b) Argue the correctness of your algorithm. [5 points]

   (c) Prove an upper bound on the time complexity of your algorithm. [5 points]

2. [**75 points**] You're in charge of logistics at the Megacorp commerce company. A Megacorp warehouse can be represented as a directed acyclic graph $G = (V, E)$, where each node represents a zone in the warehouse. All edges have a weight $w : E \to \mathbb{R}^+$, and the warehouse contains a special delivery zone $s \in V$ (s is reachable from all $v \in V/\{s\}$) (Fig 2).

   Each node except s is staffed by a fleet of delivery drones parametised by a rate r and cooldown-period c. A drone from zone u is denoted as $d_u = (r_u, c_u)$; to traverse an edge of weight l it takes $r_u \times l + c_u$ units of time.

   Your task is to find the most efficient delivery-path from all $u \in V/\{s\}$ to s.

   - A delivery-path from node u begins with drone d_u.
   - At all intermediate nodes v, the drone entering v may exit v; or alternatively it may be swapped out for d_v (a drone from node v).

   Formally, a delivery-path is defined by a sequence of edges $\{(u_1, v_1), (u_2, v_2), \ldots, (u_k, s)\}$ and drones $\{d_1, d_2, \ldots, d_k\}$ where $d_i = d_{i-1}$ or $d_i = d_{u_i}$. The cost of the path is equal to the total time it takes for delivery, i.e $\sum_{i \in 1:k} w(u_i, v_i) \times r_i + c_i$.

   You may assume that there is an infinite number of drones for each zone, i.e. the optimal $i \to s$ delivery-path does not depend on your choice of drones for any other delivery-path.

(a) Design a dynamic programming algorithm to determine the *cost* of an optimal delivery-path from each node. First derive and justify a recurrence relation and your base cases, then turn it into a bottom-up solution. For full marks, your implementation should run in $O(nm)$ time. [35 points]

(b) Argue the correctness of your algorithm. [10 points]

(c) Prove an upper bound on the time complexity of your algorithm. [10 points]

(d) Implement your algorithm (in Ed) and test it on the the given input instances. Each instance is given in a text file using the following format (where n is the number of nodes and m is the number of edges):

$\texttt{n}$

$\texttt{u}_0\ \texttt{r}_0\ \texttt{c}_0$

...

$\texttt{u}_{n-2}\ \texttt{r}_{n-2}\ \texttt{c}_{n-2}$

s

$\texttt{m}$

$\texttt{u}_0\ \texttt{v}_0\ \texttt{w}_0$

...

$\texttt{u}_{m-1}\ \texttt{v}_{m-1}\ \texttt{w}_{m-1}$

- `s` is the label for the shipping node.
- $\texttt{u}_\texttt{i}$ indicates the label for node i; $\texttt{r}_\texttt{i}$ $\texttt{c}_\texttt{i}$ corresponds to the rate and cooldown for that node's drone.
- $\texttt{u}_\texttt{i}$ $\texttt{v}_\texttt{i}$ $\texttt{w}_\texttt{i}$ corresponds to an edge from u_i to v_i with weight w_i.

For each node in ascending order, output the *cost* of the optimal delivery-path to `stdout`. You do not need to describe or analyse the time it takes to sort and output your solution. [20 points].

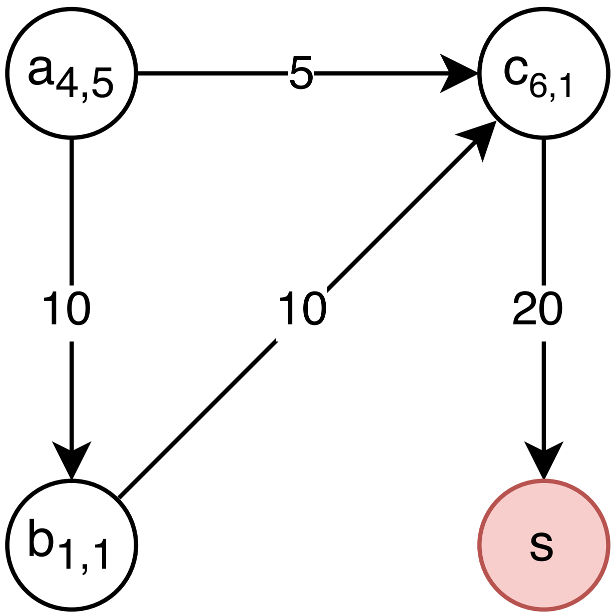

Figure 2: The rate and cool-down of a node's delivery drone is shown in subscript.

- The optimal $a \to s$ delivery paths traverses edges $(a, b), (b, c), (c, s)$ and uses drones d_a, d_b, d_b. The total cost is $(40 + 5) + (10 + 1) + (20 + 1) = 77$.
- The optimal $b \to s$ delivery paths traverses edges $(b, c), (c, s)$ and uses drones d_b, d_b. The total cost is $(10 + 1) + (20 + 1) = 32$.
- The optimal $c \to s$ delivery paths traverses the edge (c, s) and uses drone d_c. The total cost is $(120 + 1) = 121$.