

**Exercise 8.2** The following problem (and more elaborate extensions) appear in reinforcement learning. Let  $G = (V, E)$  be a directed graph and let the edges be annotated by positive edge weights  $r : E \rightarrow \mathbb{R}_{>0}$ . We think of the vertices as states of some device under our control, and the edge weights  $r(e)$  as rewards obtained by traversing the edge  $e$  as follows. Let  $s \in V$  be a fixed starting vertex / state.

**Exercise 8.2.1.** Given an integer  $k \leq n$ , the goal is to compute a walk of length at most  $k$  maximizing the sum of rewards along that walk. (If you repeat an edge, you get the same reward each time you repeat the edge.) Design and analyze an algorithm for this problem.

*Solution.* solution

□

**Exercise 8.2.2.** Here we also incorporate a *discount rate*. Let  $\alpha \in (0, 1)$  be given. Given a walk with edges  $e_1, \dots, e_k$ , the *discounted total reward* of the walk is given by

$$r(e_1) + \alpha r(e_2) + \dots + \alpha^{k-1} r(e_k).$$

The idea is that if we think of each edge traversal as also taking a unit of time, then the rewards attained far off in the future are perceived to be worth less than the rewards attained now and in the short term. Given an integer  $k \leq n$ , the goal is to compute a walk of length at most  $k$  maximizing the discounted total reward of the walk. Design and analyze an algorithm for this problem.

*Solution.*

□