[20 pts] Augmentation

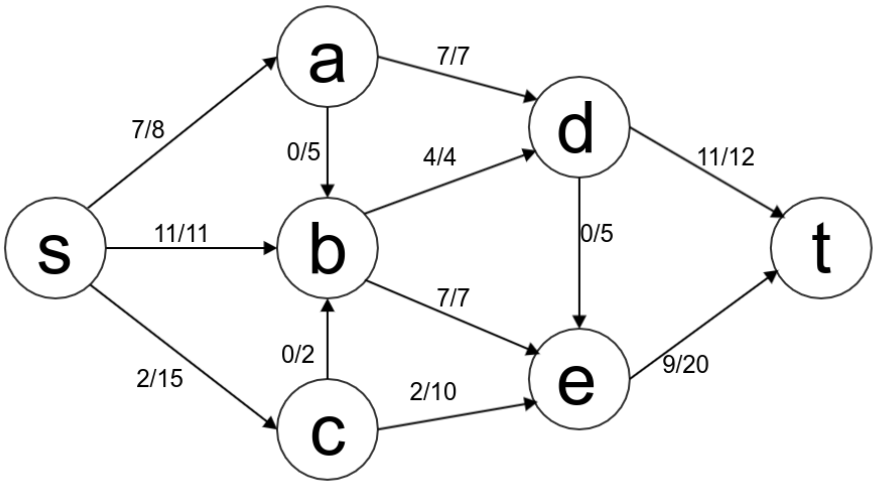
You are in charge of the salary database for Meancorp, which stores all employee salaries in a 2-3 tree ordered by salary. Meancorp compiles regular reports to the Department of Fairness about the salary for low-income employees in the firm. You are asked to implement a new database operation AVERAGE(x) which returns the average salary of all employees whose salary is at most x.

(a) [10 pts] What extra information needs to be stored at each node? Describe how to answer an AVERAGE(x) query in O(lg n) time using this extra information.

(b) [10 pts] Describe how to modify INSERT to maintain this information. Briefly justify that the worst-case running time for INSERT remains O(lg n).

[20 pts] Network Flow

Consider the following flow network G and initial flow f, where the first number on an edge indicates the current flow on that edge and the second number on an edge indicates the capacity of that edge. The source and sink nodes are represented with s and t respectively. We will perform one iteration of the Edmonds-Karp algorithm.



(a) [10 pts] Draw the residual graph G_f of G with respect to f.

(b) [5 pts] Perform the path augmentation. What is the value of the resulting flow?

(c) [5 pts] What is the value of the maximum-flow? Why?

[20 pts] NP Reduction

In the bin-packing problem we are given a set of n objects and k bins, where the size s_i of the ith object satisfies $0 < s_i < 1$ and each bin has capacity 1. Each bin can hold any subset of the objects whose total size does not exceed 1. In the decision version of this problem we are asked whether the n given objects will fit k bins. We will prove that bin-packing is NP-complete using the following steps:

1. Show that bin-packing is in NP.
   (a) Polynomial size certificate.
   (b) Polynomial time verification.
2. Show that bin-packing is NP-hard by reduction from a known NP-hard problem.
   (a) Find a polynomial time algorithm to convert the input of a known NP-hard problem A to an equivalent input for bin-packing.
   (b) Show that a yes input for A corresponds to a yes input for bin-packing.
   (c) Show that a yes input for bin-packing corresponds to a yes input for A.

(a) [2 pts] Show that you can describe a solution to a bin-packing problem using a polynomial size certificate (i.e. a formal description of the solution).

(b) [2 pts] Show that you can verify a solution to a bin-packing problem using a polynomial time algorithm.

(c) [10 pts] Give a polynomial time algorithm to convert an input for the subset-sum problem to an equivalent input to the bin-packing problem. Subset-sum is a known NP-complete problem where given a set of positive integers S and a positive integer target value t we need to decide whether there exists a subset of S that adds up exactly to the target value t. (You can use another known NP-hard problem instead of subset-sum if you wish).

(d) [3 pts] Suppose we have a "yes" instance for subset-sum, i.e. the set S does have a subset that adds up to t. Show that your conversion gives a "yes" instance of the bin-packing problem where the n objects will fit the k bins.

(e) [3 pts] Conversely suppose that we have a "yes" instance of the bin-packing problem where the given n objects fit k bins. Show that the corresponding subset-sum problem also has a "yes" answer.

[20 pts] Rest days

A worker is offered n days of work and is told that he can choose which days he wants to work. If he chooses to work on day i he receives an income of v_i but cannot work the next r_i days in order to rest.

(a) [10 pts] Given r_i and v_i for $i = 1 \dots n$, find an algorithm that can compute the maximum income the worker can receive.

(b) [5 pts] How can you update your algorithm to also output *which days* the worker must choose to work to get the maximum income?

(c) [5 pts] Analyze the time and space complexity of your algorithm.

[20 pts] The End of Year Bonus

Peach Inc. had an outstanding financial performance in 2021 and has decided to give a bonus to their n employees. The company has evaluated each employee and rated their performance by an integer; to be more specific, R_i is the score of the i-th employee. Additionally, the bonuses are given in the form of gift boxes, and each employee will receive at least one box. All the employees sit in a line, and if two employees sit next to each other, then the one with a higher rate must get more boxes. If two employees sitting next to each other have equal scores, they are allowed to have different number of boxes. The boss wants to give out the minimum number of boxes that satisfy these constraints.

For example, if $n = 3, R = [1, 2, 2]$, the optimal box distribution is $[1, 2, 1]$ and the cost will be 4 boxes.

If $n = 4, R = [1, 2, 3, 4]$, the optimal box distribution is $[1, 2, 3, 4]$ and the cost will be 10 boxes.

(a) [10 pts] Given n and R, find the minimum number of boxes that the company needs to buy. Describe your algorithm in detail with pseudo-code.

(b) [10 pts] What is the time and space complexity of your algorithm? Please provide a detailed calculation.