Orlin's algorithm computes maximum flows in $O(VE)$ time.
You may use this algorithm as a black box.

1. Suppose you are given a flow network G with ***integer*** edge capacities and an ***integer*** maximum flow f^* in G. Describe algorithms for the following operations:

   (a) INCREMENT(e): Increase the capacity of edge e by 1 and update the maximum flow.

   (b) DECREMENT(e): Decrease the capacity of edge e by 1 and update the maximum flow.

   Both algorithms should modify f^* so that it is still a maximum flow, more quickly than recomputing a maximum flow from scratch.

2. Suppose we are given a set of boxes, each specified by its height, width, and depth in centimeters. All three dimensions of each box lie strictly between 25cm and 50cm; box dimensions are *not* necessarily integers. As you might expect, one box can be placed inside another box if, possibly after rotating one or both boxes, each dimension of the first box is smaller than the corresponding dimension of the second box. Boxes can be nested recursively.

   Call a box is *visible* if it is not inside another box. Describe and analyze an algorithm to nest the boxes so that the number of visible boxes is as small as possible.

3. The University of Southern North Dakota at Hoople has hired you to write an algorithm to schedule their final exams. Each semester, USNDH offers n different classes. There are r different rooms on campus and t different time slots in which exams can be offered. You are given two arrays $E[1..n]$ and $S[1..r]$, where $E[i]$ is the number of students enrolled in the ith class, and $S[j]$ is the number of seats in the jth room. At most one final exam can be held in each room during each time slot. Class i can hold its final exam in room j only if $E[i] < S[j]$.

   Describe and analyze an efficient algorithm to assign a room and a time slot to each class (or report correctly that no such assignment is possible).