Greedy and Exchange Argument

Problem 1. At your summer job, you are made in charge of storing books on shelves in a library. Consider n books where each book i has some thickness t_i. The order of storing the books is fixed by the cataloging system and so you cannot rearrange them. Every shelf in the library has a fixed length L. Suppose any book fits on any shelf by itself. Design an algorithm to store all the n books using the fewest number of shelves possible. Show that your algorithm is correct.

Solution. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Problem 2. Let X be a set of n intervals on the real line. We say that a set P of points stabs X (or is *the stabbing set*) if every interval in X contains at least one point in P. See Figure 2. In this and the next question, we will design and analyze an efficient algorithm to compute minimum set of points that stabs X. Assume that your input consists of pairs (ℓ_i, r_i), representing the left and right endpoints of ith interval in X, for $1 \le i \le n$.
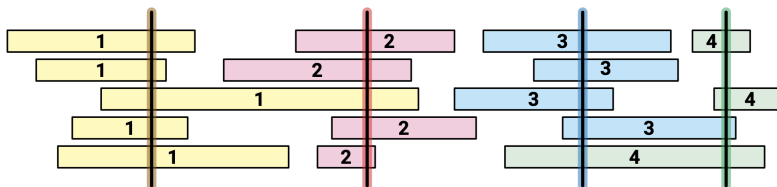

Figure 1: A set of intervals stabbed by four points. From Jeff Erickson's Algorithms book.

(a) *Structure of optimal.* If we were to choose stabbing points anywhere on the intervals, the possibilities would be endless. So first, we prove a structural property about any optimal solution to the problem, as this will guide our algorithm design. Show that for any set of intervals, there exists an optimal solution (a stabbing set of minimum size) where every stabbing point is the right endpoint r_i of some interval. This means that without loss of generality (WLOG), we can restrict ourselves to stabbing sets that satisfy this property.

Solution. □

(b) Consider the following greedy strategy: sort the intervals in increasing order of their right endpoints. Take the first interval from this list, and add its right endpoint to the stabbing set. Remove all intervals that contain this point. Repeat until no intervals remain. Clearly, this produces a valid stabbing set, but is it optimal?

Prove that the above greedy strategy is optimal, that is, it produces a stabbing set of minimum size. (Use part (a) and an exchange argument.)

Solution. □

(c) Analyze the running time of the greedy strategy and show that it can be implemented in $O(n \log n)$ time.

Solution. □

Greedy Algorithms on Graphs

Problem 3. Are the following statements True of False? You must justify your choice with a brief explanation or a counterexample.

(a) The minimum spanning tree of a graph includes the minimum-weight edge in *every* cycle.

Solution. □

(b) Kruskal's algorithm for finding minimum spanning trees works for negative edge weights.

Solution. □

(c) Dijkstra's shortest path algorithm works even when the graph has negative edge weights.

Solution. □

Problem 4. It is often the case that updating the output of an algorithm when the input changes is easier than computing it from scratch. One such example is that of dynamically changing edge weights in a graph. This problem asks you to consider how you would maintain a minimum-cost spanning tree in such an environment. Suppose you are given a graph G with weighted edges and a minimum spanning tree T of G.

Describe and analyze an algorithm to update the minimum spanning tree when the weight of a single edge e is decreased.[1] The input to your algorithms is the edge e and its new weight and your algorithms should modify T so that it is still a minimum spanning tree. Remember to prove correctness and analyze the running time of your algorithm.

Assume that all edge costs are distinct. *Hint: Consider $e \in T$ and $e \notin T$ separately and use the properties that relate MST, cycles, and cuts.*

Solution. □

---

[1]Food for thought: what if we increase the weight of a single edge? What do we update the MST then?

Problem 5. A *feedback edge set* in a graph G is a subset F of the edges of G such that every cycle in G contains at least one edge in F. In other words, removing every edge in F makes the graph G acyclic. Notice that the set of all the edges of the graph trivially forms a feedback edge set.

(a) Let a feedback edge set F be *minimal* if it does not have another feedback edge set as its subset, that is, there is no set $F' \subset F$ such that F' is also a feedback edge set.[2] Show that if T is a spanning tree of G, then $F = E - T$ is a minimal feedback edge set.

   Solution. □

(b) Show that if F is a minimal feedback edge set of G, then the edges $E - F$ form a spanning tree T of G.

   Solution. □

(c) Use parts (a) and (b) to give an algorithm to compute the *minimum-weight* feedback-edge set of a given edge-weighted graph. Argue the correctness and runtime of your algorithm. (Assume that G is connected and all its edge weights are distinct and positive.)

   Solution. □

---

[2]Note that this has nothing to do with edge costs!

Acknowledgment

This is where you cite your sources and collaborators.