Some sample problems

Note these problems are not necessarily representative of the overall content of the final, but are some extra ones to practice if you'd like.

In addition, I've chosen "more interesting" problems here (particularly the longer ones), so they're likely a bit more challenging than the ones on the final will be.

Prim's and Bellman-Ford algorithms are greedy algorithms. True or false?

Computing the median of n elements takes $\Omega(n \log n)$ time for any algorithm working in the comparison-based model. True or false?

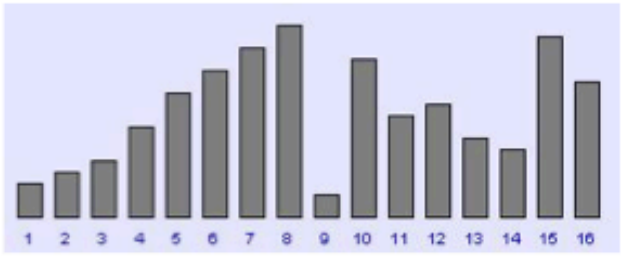
Consider a modification to the deterministic quicksort algorithm in which, instead of using the first element as a pivot, we use the median element that we find using deterministic order statistics algorithm described in class. What's the worst-case running time of the resulting algorithm?

Let S be a set of n integers. There exists a (possibly randomized) data structure for S so that determining whether an integer x belongs to S can be performed in $O(1)$ time in the (possibly expected) worst case. True or False?

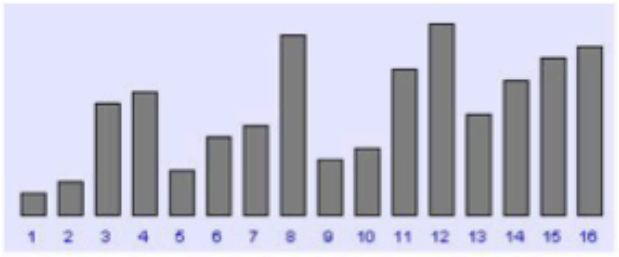
(36) You are given an array of n intervals $I_1, \ldots, I_n$. For a collection C of non-overlapping intervals, let S_C be the sum of the lengths of its elements. Give an algorithm that returns the maximum value of S_C. What is the running time of your algorithm? You get full credit for a $O(n \log n)$ time algorithm, partial credit for a polynomial-time algorithm.

(8 points) Sorting algorithms: The pictures below each depict a sorting algorithm processing on an array of 16 elements about half of the way through. The heights of the bars indicate the relative values. A fully sorted array would have the bars in increasing size from left to right. Label each figure with the appropriate search algorithm: selection, insertion, quicksort or mergesort. Each algorithm should be used once as a label.

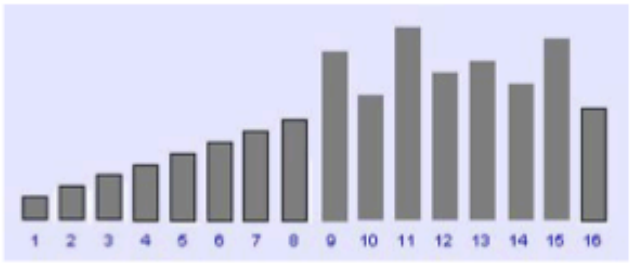
_____



_____



_____



_____

(24) Your friend has n mobile robots that can communicate with each other over wireless links. The wireless links have limited range, and thus a robot might be able to directly communicate only with a subset of other robots, i.e. with robots that are close enough to him.

Consider a graph G where nodes correspond to robots and there is an edge vw if robots v and w are close enough to each other and hence can communicate directly. As long as G is connected, it is possible to route traffic between any two robots by forwarding the packets over multiple links.

Your friend claims that if he constrains the movement of robots to ensure that at each time each robot can communicate (directly, i.e without relying on forwarding by others) with at least $n/2$ other robots, the communication graph is connected and hence all robots can communicate with each other using an appropriate routing algorithm.

Prove or disprove your friend's claim.

(5 points) You have three containers with sizes 10 pints, 7 pints and 4 pints. The 7 and 4 pint containers start out full of water, but the 10 pint container is empty. You are allowed to pour the contents of one container into another until either a) you source container is empty or b) the destination container is full. We want to know if there is a sequence of pourings that leaves exactly 2-pints in either the 7 or 4 pint container.

(a) Model this as a graph problem. Describe exactly how to construct the graph and what question about the graph needs to be answered.

(b) What algorithm should be applied to this problem?