**CS 430 – FALL 2017**

**INTRODUCTION TO ALGORITHMS HOMEWORK #3  
DUE Fri Oct 6, 11:25am**

1.

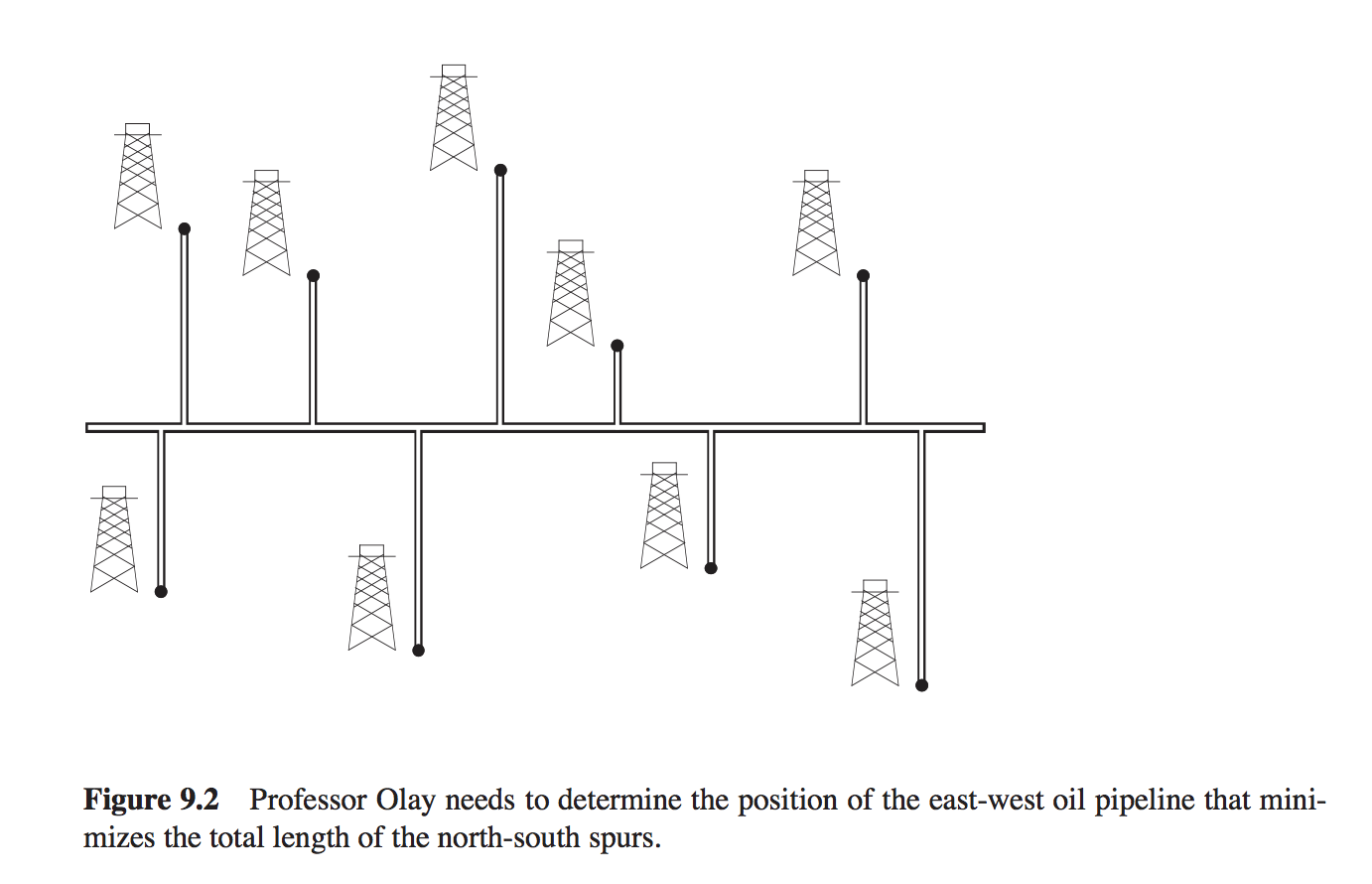
***9.1-2***

Prove the lower bound of [3n/2] - 2 comparisons in the worst case to find both the maximum and minimum of n numbers. (*Hint:* Consider how many numbers are potentially either the maximum or minimum, and investigate how a comparison affects these counts.)

2.

***9.3-9***

Professor Olay is consulting for an oil company, which is planning a large pipeline running east to west through an oil field of n wells. The company wants to connect a spur pipeline from each well directly to the main pipeline along a shortest route (either north or south), as shown in Figure 9.2. Given the x- and y-coordinates of the wells, how should the professor pick the optimal location of the main pipeline, which would be the one that minimizes the total length of the spurs? Show how to determine the optimal location in linear time.



3.

***9-1 Largest*** i ***numbers in sorted order***Given a set of n numbers, we wish to find the i largest in sorted order using a comparison-based algorithm. Find the algorithm that implements each of the following methods with the best asymptotic worst-case running time, and analyze the running times of the algorithms in terms of n and i.

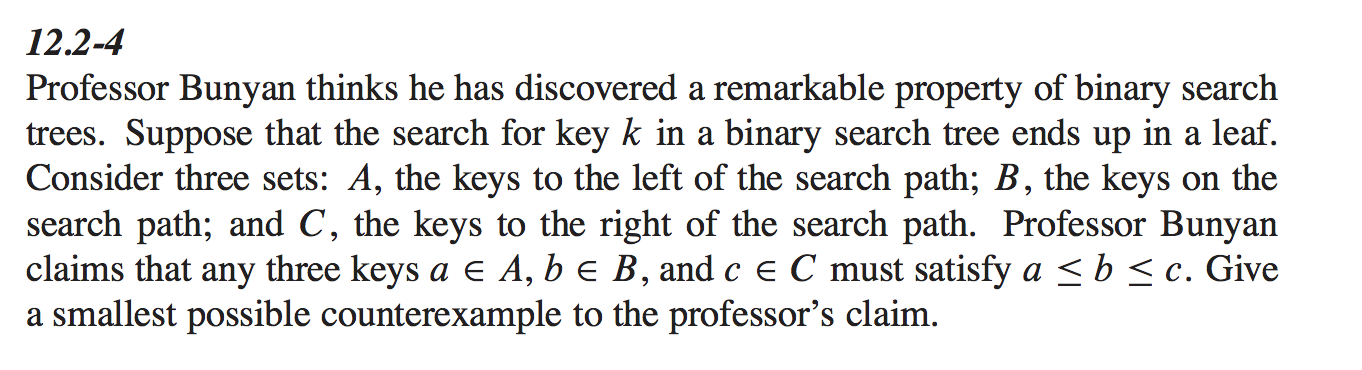
1. ***Sort the numbers, and list the i largest.***
2. ***Build a max-priority queue from the numbers, and call EXTRACT-MAX i times.***
3. ***Use an order-statistic algorithm to find the ith largest number, partition around that number, and sort the i largest numbers.***

4.

***12.1-5***

Argue that since sorting n elements takes Ω(nlgn) time in the worst case in the comparison model, any comparison-based algorithm for constructing a binary search tree from an arbitrary list of n elements takes Ω(nlgn) time in the worst case.

5.



6.

***12.2-5***

Show that if a node in a binary search tree has two children, then its successor has no left child and its predecessor has no right child.

7.

(3 points) Given an arbitrary binary tree T with integer keys stored at the nodes, design an efficient algorithm which determines whether or not T is a binary search tree. What is the time complexity of your algorithm?