CS302 - Problem Set 6

1. `Selection`(A, k) finds the k^{th} smallest element of an array A. For example, if $A = [3, 11, 20, 6, 7, 38]$ and $k = 3$, then `Selection`(A, k) returns 7 because 7 is the value of the 3rd smallest element. Here is pseudocode for a divide and conquer algorithm for `Selection`.

   Note that in the following pseudocode I index A starting at 1, not 0. $A[s, f]$ is the subarray of A from index s to f inclusive of both s and f.

Algorithm 1: `Selection`(A, k)

Input : An array A of n integers indexed starting at 1 and an integer $k \in \{1, \ldots, n\}$

Output: The kth smallest element of A.

```
1  piv ← ChoosePivot(A);
2  Partition(A, piv);
3  Let p be the index of the pivot after Partition;
4  if p = k then
5  |  return A[p];
6  end
7  if p < k then
   |  // Everything larger than the pivot, including the kth
   |     element, is to the right of p
8  |  return Selection(A[p + 1 : n], k − p);
9  else
   |  // Everything smaller than the pivot, including the kth
   |     element, is to the left of p
10 |  return Selection(A[1 : p − 1], k);
11 end
```

   Now consider the case that `ChoosePivot` chooses a pivot uniformly at randomly (each index is chosen with equal probability). Consider the random variable X_{ij}, for $i < j$, which is the number of times the i^{th} and j^{th} smallest elements of A are compared at some point in the algorithm.

   (a) What is the sample space of this problem?

   (b) Explain why X_{ij} is an indicator random variable.

(c) What is the probability of z_i and z_j being compared if we are trying to find the kth order statistic, and $k < i, j$?

(d) What is the probability of z_i and z_j being compared if we are trying to find the kth order statistic, and $i, j < k$?

(e) What is the probability of z_i and z_j being compared if we are trying to find the kth order statistic, and $i \leq k \leq j$?

(f) Use linearity of expectation and properties of the expectation value of indicator random variables to create an explicit expression involving sums that gives the average number of comparisons done over the course of the algorithm.

(g) **Challenge** Analyze the sums from the previous part to get $O(n)$.

2. Finish the proof of Lemma 1 from the proof of Huffman's algorithm from class:

Lemma: Given an alphabet Σ and probabilities $p(i)$ for each $i \in \Sigma$, let a and b be the letters with the smallest probabilities. Then there is an optimal tree (a tree corresponding to a code with smallest average letter length) where a and b are siblings.

Proof: Suppose for contradiction that there are no optimal trees where a and b are siblings. Let T' be an optimal tree where a, b are not siblings. Let $x, y \in \Sigma$ be letters that are sibling leaves in T' that also have maximum depth. (Note that at least one of x or y must not equal a or b, because otherwise a and b would be siblings. Also there must be a pair of siblings at maximum depth because if there is only one leaf at maximum depth, then you can relocate that symbol to be at its parent node while maintaining a prefix free tree, which would then result in a tree with smaller average length than T', contradicting the fact that T' is optimal.)

Let d be the depth of x and y in T', and let $d(i)$ be the depth of any other letter i in T'.

Consider the tree T'_{ex} that we get by exchanging the positions of a and x and exchanging the positions of b and y, while leaving all other letters in the same positions. Then ...

3. Suppose I would like to give you more flexibility on your exams, so I give you some large number (let's call it M) of problems, where the ith problem is worth p_i points, and I give partial credit. The time I give you to take the exam is not sufficient to solve all of the problems, so each student might solve a different subset of problems and might get different grades on each problem. I would like to give a good grade to a student who does sufficiently well on a sufficient number of questions, and give a worse grade to a student who just gets partial credit on a lot of problems. Then grading problem is: given a list $v = (v_1, v_2, \ldots, v_M)$, where v_i is the number of points (out of p_i) that a student has scored on question i, how do I determine their grade, a number between 0 and 100?

(a) Reduce the grading problem to the knapsack problem.

(b) Is your reduction a polynomial reduction? Explain.

(c) Do you feel this is an equitable form of grading? Do you feel it is a form of grading that accurately captures the amount of learning a student accomplishes? Would you like to have professors use this grading scheme to determine your grade? Why or why not?