CS302 - Problem Set 5

1. Suppose you are hosting a music festival, and are trying to decide which bands should play in the prime-time slot. You randomly surveyed 100 of the registered attendees to have them rank the bands who will be at the festival. You have an unlimited number of stages, so you can have many different bands play at the same time. You would like to choose a subset of bands to play during the prime time spot such that (1) most attendees will be excited about at least one of the bands playing in the primetime slot, and also (2) most attendees will *not* be excited about *more than one* of the bands playing in the primetime slot (since these people would really be bummed that they have to make a choice and only see one of the bands that they paid to see). For example, suppose some of the attendees like jazz, and some like metal, but not many people like both jazz and metal. Then you could have the most popular jazz band and the most popular metal band play the prime time slot. This would be better than having both of your jazz bands play during the prime time slot, even if they are both more popular overall than any of the metal bands, because then the metal fans would not have anyone to watch in the prime time slot, and the jazz fans would be annoyed that they would have to pick one of the bands to watch and couldn't see both.

   (a) Describe a possible reduction from the Music Festival problem to the Max Weight Independent Set (MWIS) problem. The input to the Music Festival problem is a list of bands and the survey data from the 100 attendees, and the output should be a set of bands to play in the prime time slot. (There is more than one way to do the reduction - you should make choices based on what you think will be best for your music festival.)

   (b) Explain why the reduction you described gives a good solution to the Music Festival problem. Do you have any ethical or societal concerns?

   (c) Is the Music Festival problem polynomial time reducible to MWIS?

   (d) Now that you've figured out who should perform in the prime time slot, how would you pick which bands should perform in the second best time slot?

2. Suppose you are living in a country that has a coin worth 6 cents, a coin worth 4 cents, and a coin worth 1 cent. This society also values having small wallets, so whenever they make change, they want to use as few coins as possible. For example, if they need to make 8 cents worth of change, they would rather return two 4-cent coins (2 total coins), rather than a 6-cent coin, and two 1-cent coins, (3 total coins). You are tasked with creating a program for cash registers that will help make change using the least number of coins.

- **Input:** An integer n
- **Output:** A list $M = (m_6, m_4, m_1)$ of non-negative integers such that $n = 6m_6 + 4m_4 + m_1$ and $C(M) = m_6 + m_4 + m_1$ is minimized. (Here m_i represents the number of i-cent coins that should be returned, and $C(M)$ is the number of coins returned; the output should make the correct change for n while using the fewest number of coins.)

(a) **Challenge:** Design a dynamic programming algorithm to solve this problem by following the same process we went through in class for MWIS. (If you would like guidance continue to the next question).

(b) To design a dynamic programming algorithm, we divided up the original problem into a series of subproblems, each slighly smaller than the first, and picked as our recurrence object the optimal strategy/solution for each suproblem. (For MWIS on a line, the subproblems were the graphs G_i and the recurrence object was S_i, the MWIS on G_i). Next we thought about options for a "final choice" for our recurrence object that allowed us to come up with a recurrence for subproblems (for MWIS on a line, we considered $v_n \in S_n$ and $v_n \notin S_n$). What subproblems, recurrence objects, and final choices should we use for our coin problem?

(c) Create a recurrence relation based on each final choice. Write a brief explanation describing why your recurrence relation is correct.

(d) What are the base cases of your recurrence relation?

(e) Explain why the recurrence relation you determined makes this problem an appropriate candidate for dynamic programming, rather than a usual recursive algorithm.

(f) Based on your recurrence relation from the previous part, create a recurrence relation for the optimal objective function value of subproblems rather than the optimal strategy for subproblems.

(g) Using your recurrence relations, fill in the pseudocode below:

```
Algorithm 1: CoinCounter($n$)
Input  : $n \geq 1$
Output: $(m_6, m_4, m_1)$, the optimal number of 6-, 4- and
        1-cent coins to make $n$ cents.
// Fill in array $A$ such that $A[i]$ is the minimum
   number of coins needed to create $n$ cents
1 $A[0] \leftarrow 0$;



// Work backwards through $A$ to get values for
   $(m_6, m_4, m_1)$, the optimal number of 6-, 4- and
   1-cent coins to make $n$ cents.
2 $m_6 \leftarrow 0$;
3 $m_4 \leftarrow 0$;
4 $m_1 \leftarrow 0$;
5 $k \leftarrow n$;
6 while $k \geq 1$ do
  |
  |
  |
7 end
8 return $(m_6, m_4, m_1)$;
```

(h) What is the runtime of your algorithm?

3. 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. 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 2: `Selection`(A, k)

```
Input  : An array $A$ of $n$ integers indexed starting at 1 and an integer
         $k \in \{1, \ldots, n\}$
Output: The $k$th smallest element of $A$.
1  $piv \leftarrow$ 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 $k$th
          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 $k$th
          element, is to the left of $p$
10     return Selection$(A[1 : p-1], k)$;
11 end
```

(a) What is the worst case runtime of this algorithm if `ChoosePivot` somehow always chooses the pivot to be the median of the array, and the array is originally size n?

(b) Create and analyze a recurrence relation for the runtime of this algorithm if `ChoosePivot` somehow always chooses the pivot to be the smallest element the array, and the array is originally size n.

(c) Now consider the case that `ChoosePivot` chooses a pivot uniformly at randomly. 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.

   i. What is the sample space of this problem?

   ii. Explain why X_{ij} is an indicator random variable.

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

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

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

   vi. 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.

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

4. Tying into our learning about QuickSort, we will discuss one of the most ubiquitious sorting algorithms in use today: internet search engine sorting algorithms, particularly Google's PageRank. (Google Search does not use QuickSort! This is a thematic rather than literal tie-in.) People rarely go beyond the first page of search results, so the way the Google sorts and prioritizes results has a large effect on the type of information users are exposed to.

   Note: the following pieces discuss racist and sexual images/text, but do not contain explicit content. Either watch Dr. Safiya Noble's Ted Talk or read her piece in TIME (or both!). Write a brief response/reflection on at least some of the following questions. We will discuss in class.

   (a) Based on the examples Dr. Noble brings up, what type of user/content is Google prioritizing in its top search results? How is this at odds with how Google attempts to portray its sorting of content, and how many people perceive the objectivity of its results? (Take a look at the design of the Google Homepage - what signals is the design trying to send the user about its results?)

   (b) What groups of people are most harmed/receive the most benefit from the way that Google chooses to sort its search results? How are these effects exacerbated when most people believe Google search results to be accurate/objective?

   (c) How often do you question whether the top page Google results are the most relevant for your search? Are there searches that you typically "Google" that you might now think about asking of other resources? What types of searches and what other types of resources?

   (d) How does Google make money? How can websites with access to money/power influence search results? How do these factors tie into the search results Google prioritizes in its algorithm? Why is this problematic?