Homework # 1 for Introduction to Algorithms (Spring of 2021)
Due date: April 11
Instructor: Kyuseok Shim

1. (Algorithm Analysis) Professor Shim thinks he has a great new sorting algorithm. Here is the pseudocode for his proposed algorithm. The input is a list A[1...n] of n numbers, where n is a power of 2.

```
ShimSort(A)
begin
1.  if length(A) = 1
2.     return A
3.  ShimSort(A[1..n/2])
4.  ShimSort(A[n/2+1..n])
5.  for i = 1 to n/2
6.     if A[i] > A[i + n/2]
7.        Swap(A[i],A[i + n/2])
8.  ShimSort(A[1..n/2])
9.  ShimSort(A[n/2+1..n])
end
```

   - Analyze the running time of Professor Shim's sorting algorithm. (State and solve the recurrence.) [10 pts.]
   - Does the algorithm sort correctly? If Yes, argue why. If No, give an example for which the above algorithm does not sort correctly. [10 pts.]

2. (Divide and Conquer) You have just been hired as the quality-control engineer for a company that makes coins. The coins must have identical weight. You are given a set of n coins, and are told that exactly 1 of the n coins is too light (but you do not know which coin). The other n-1 coins have identical weights. Your task is to develop an efficient test procedure to determine which of the n coins is defective. To do this test you have a scale. For each measurement you place some of the coins on the left side of the scale and some of the coins on the right side. The scale indicates either (1) the left side is heavier, (2) the right side is heavier, or (3) both subsets have the same weight. It does not indicate how much heavier or lighter. You may assume n is a power of 2, 3 or 4 if it simplifies your algorithm, but state your assumption.

   - Present a method to determine the defective coin using $O(\log n)$ scale measurements. You should clearly state your method, state the recurrence for the number of measurements, and solve the recurrence. [10 pts.]
   - Suppose you are given n weighted coins, with exactly 1 defective coin, but we do not know whether the defective coin is lighter or heavier than the $n-1$ other coins. Present a

method to determine the defective coin in this setting using $O(\log n)$ scale measurements. [10 pts.]

3. (Selection in Worst-case Linear Time) T he recurrence for the worst-case running time T(n) of the algorithm SELECT in the text book is $T(n) \leq T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n)$. If this algorithm SELECT in step 1, divides the n elements of input array into $\lfloor n/7 \rfloor$ groups of 7 elements each and at most one group made up of the remaining (n mod 7) elements, then how can the recurrence of T(n) be expressed? Show the running time of that recurrence equation by substitution method. [15 pts.]

4. (Dynamic programming) There is a 2-row, n-column matrix of points w(i,j). We want to choose the points to maximize the total sum of points with below constraints. [30 pts.]

   - You should choose only one point from each column of matrix.
   - You can choose the points consecutively up to two in the same row of matrix.

-4	5	11	14	5	6	-4	8	14	19
6	12	-3	12	-5	-11	1	-6	12	7

   (a) Give the recursive formula of the problem. [10 pts.]

   (b) Show the optimal substructure property. [5 pts.]

   (c) Show the overlapping subproblem property. [5 pts.]

   (d) Write down a pseudocode for the above problem as a top-down, recursive manner. [5 pts.]

   (e) Give the time and space complexities of your algorithm. [5 pts.]

5. (Dynamic Programming) Here we look at a problem from computational biology. You can think of a DNA sequence as sequence of the characters `a`, `c`, `g`, `t`. Suppose you are given DNA sequences D_1 of n_1 characters and DNA sequence D_2 of n_2 characters. You might want to know if these sequences appear to be from the same object. However, in obtaining the sequences, laboratory errors could cause reversed, repeated or missing characters. This leads to the following sequence alignment problem. An alignment is defined by inserting any number of spaces in D_1 and D_2 so that the resulting strings D_1' and D_2' both have the same length (with the spaces included as part of the sequence). Each character of D_1' (including each space as a character) has a corresponding character (matching or non-matching) in the same position in D_2. For a particular alignment A we say $cost(A)$ is the number of mismatches (where you can think of a space as just another character and hence a space matches a space but does not match one of the other 4 characters). To be sure that this problem is clear, suppose that D_1 is `ctatg` and D_2 is `ttaagc`. One possible alignment is given by:

```
ct at g
 tta agc
```

   In the above both D_1' and D_2' have length 8. The cost is 5. (There are mismatches in position 1, 3, 5, 6 and 8). Give the most efficient dynamic programming algorithm you can (analyzed as a function of n_1 and n_2) to compute the alignment of minimum cost. [25 pts.]

Show recursive formula of dynamic programming (10 pts.), the pseudocode of your algorithm (10 pts.) and time complexity (5 pts.) only!]