CS 6363: Algorithms – Fall 2019
Homework #2 – Due: Sept. 23
Professor D.T. Huynh

Problem #1. Given an array of integers $A[1..n]$ such that $|A[i] - A[i+1]| \leq 1$ for all $i = 1, \ldots, n-1$. (That is the values of adjacent elements differ by at most 1.) Let $x = A[1]$ and $y = A[n]$, and assume that $x < y$. Design an efficient search algorithm to find j such that $A[j] = z$ for a given value z with $x \leq z \leq y$. Estimate the number of comparisons required by your algorithm (it should be $O(logn)$). Using the decision tree method, show that your algorithm is optimal.

Problem #2. You have n coins that are supposed to be gold coins of the same weight, but you know that one coin is fake and weighs less than the others. There is available a balance scale so that you can put any number of coins on each side of the scale at one time, and it will tell you whether the two sides weigh the same, or which side weigh less than the other. Outline an *efficient* algorithm to find the fake coin. Determine the number of weighings as a function of n using the O notation.

Problem #3. Let $A[1..n]$ be an array of n elements. An element x is called a *majority element* in $A[1..n]$ if $Card(\{ i \mid A[i] = x \}) > 2n/3$. Give an $O(n)$ algorithm that decides whether $A[1..n]$ contains a majority element, and if so finds it.

Problem #4. Consider the alphabet $\Sigma = \{a, b, c\}$, and the following multiplication table: (note that this multiplication is *not* associative)

	a	b	c
a	c	b	b
b	a	b	a
c	b	a	a

Design an efficient algorithm that examines an input string $w = w_1 \ldots w_n \in \Sigma^*$, and decides whether or not it is possible to parenthesize w in such a way that the value of the resulting expression is a. For instance, if $w = bbbba$, then your algorithm should return "Yes" since $((b(bb))(ba)) = a$. (Hint. Use dynamic programming!)

Problem #5. Let X, Y, Z be strings over the alphabet Σ. Z is said to be a *shuffle* of X and Y if Z can be written as $Z = X_1Y_1X_2Y_2 \ldots X_nY_n$, where X_i and Y_j are strings in Σ^* such that $X = X_1X_2 \ldots X_n$ and $Y = Y_1Y_2 \ldots Y_n$. For example *cchocohilaptes* is a shuffle of *chocolate* and *chips*, but *chocochilatspe* is not. Devise a dynamic-programming algorithm that determines for X, Y, Z in the input whether Z is a shuffle of X and Y. Analyze the complexity of your algorithm. (Hint. Construct a Boolean table.)

Problem #6. Do Problem 15.4-5 on page 397 in [CLRS].

==

For practice purpose work on problems #15-1, #15-2, #15-6 in Chapter 15, p. 404-408 of textbook. These problems will not be graded.