EXERCISES FOR CSE202

Exercise 1. *In the problem k-sum, you are given an n-tuple of nonnegative integers $L = (x_1, \ldots, x_n) \in \mathbb{N}^n$ and a target sum $S \in \mathbb{N}$. The problem is to determine whether there exists a subset of A of $\{1, \ldots, n\}$* of cardinality k *such that*

$$\sum_{i \in A} x_i = S.$$

(1) *Show that this problem has a solution in polynomial complexity.*
(2) *For $k = 2$ give an algorithm (in pseudo-code) solving the problem in $O(n \log n)$ operations (addition and comparison of integers are assumed to have constant cost).*
(3) *Deduce an algorithm in $O(n^2)$ operations for $k = 3$ (give its pseudo-code).*

Exercise 2. *Let $\mathrm{Mul}(n)$ denote the complexity of multiplying polynomials of degree at most n in $\mathbb{K}[X]$ in terms of number of arithmetic operations on the coefficients in the field $\mathbb{K}$, assumed to satisfy $\mathrm{Mul}(n_1) + \mathrm{Mul}(n_2) \leq \mathrm{Mul}(n_1 + n_2)$ and $\mathrm{Mul}(mn) \leq m^2 \,\mathrm{Mul}(n)$.*

For simplicity, n is assumed to be a power of 2.

(1) *Given $(a_1, \ldots, a_n) \in \mathbb{K}^n$, show that the coefficients of the polynomial $(X - a_1) \cdots (X - a_n)$ can be obtained in $O(\mathrm{Mul}(n) \log n)$ operations.*
(2) *Given $P \in \mathbb{K}[X]$ of degree n show that the values of $P(a_1), \ldots, P(a_{n/2})$ can be obtained from the values taken at $(a_1, \ldots, a_{n/2})$ by the polynomial $R(X)$ remainder of the Euclidean division of P by $(X - a_1) \cdots (X - a_{n/2})$.*
(3) *Deduce a divide-and-conquer algorithm for the evaluation of P at $(a_1, \ldots, a_n)$ (give pseudo-code).*
(4) *Estimate the complexity of that algorithm.*

Exercise 3. *If comparison is cheap, but writing is expensive, then the number of* exchanges *performed by Quicksort is an important parameter in the study of the time complexity. Assume that the algorithm is applied to a permutation of $\{1, \ldots, n\}$ drawn uniformly at random.*

(1) *Show that the average number of exchanges during the first partition stage is $(n - 2)/6$.*
(2) *Find the asymptotic behaviour of the average number of exchanges during the algorithm, first writing a linear recurrence that it satisfies.*

Exercise 4. *In the Multiset Identity Problem, you are given two multiset of integers (i.e., sets with repetitions) and the problem is to decide whether they are equal or not.*

(1) *Give a first solution in $O(n \log n)$ operations.*
(2) *Find a better algorithm using hashing.*

Exercise 5. *Construct a worst-case example for the Boyer-Moore algorithm over a binary alphabet in the simple variant where only the last character shift heuristic is used (ie, the max in the loop does not use the array bms).*

Exercise 6. *Prove that the following decision problem is NP-complete. Given n knights, and a set of pairs of knights who are enemies, is it possible to arrange the knights around a round table so that two enemies do not sit side by side? [Find a reduction involving one of the NP-complete problems seen in the course.]*

Exercise 7. *The decision problem SET-COVER is defined as follows: Given a set X containing n elements, m subsets $S_1, \dots, S_m$ of X and an integer $k \leq n$, is there a set of k of the subsets that covers all the elements of X? In other words, is there a subset I of $\{1, \dots, m\}$, of cardinality k, such that for all $x \in X$, $x \in \cup_{i \in I} S_i$?*

SET-COVER is an NP-complete problem. The associated optimization problem is to find the smallest cover. We consider the following greedy algorithm for this optimization problem:

- *Pick a subset S_j that covers the largest number of elements of X;*
- *Suppress from X and from the remaining S_is the elements covered by S_j, and start again.*

Analyse this algorithm by answering the following questions:

(1) *Is this algorithm optimal?*
(2) *Prove that if an optimal solution contains k subsets, the algorithm takes at most $O(k \log n)$ subsets.*
(3) *What is the approximation ratio of this algorithm?*