COP-4534: Algorithm Techniques

Homework 5

DUE: Saturday April 21 at 11:55 PM

- Please remember that all submissions must be typeset. Handwritten submissions will NOT be accepted. These must be uploaded to SCIS moodle in PDF format only.

- Please remember to type your name on top of your submission.

1. (20 points) You have $n$ pairs of nuts and bolts (the bolts screw into the nuts) that have been dropped onto a table with the nuts on one side and the bolts on another side; however you cannot tell whether one nut is bigger than another nut or whether one bolt is bigger than another bolt. You can attempt to screw a bolt into a nut and this will tell you either that the nut matches the bolt, or that the bolt is too large or too small for the nut. Call this a nut/bolt comparison. Give a randomized algorithm that will match all nuts with their corresponding bolts with expected $O(n \log n)$ number of nut/bolt comparisons. (Hint: does this problem look similar to QuickSort?)

2. (20 points) Suppose we have a table composed of $n \times m$ cells, each having a certain number of apples. You start from the upper-left corner. At each step you can either go down or go right by one cell. Design an efficient algorithm that finds the maximum number of apples you can collect.

3. (20 points) Let the input be an $n \times m$ grid and there is a number $C(i, j)$ associated with every cell in the grid. Assume the bottom row is row 1, and the top row is row $n$. The moving rule is: from cell $(i, j)$, you can move to cell $(i + 1, j - 1)$ if $j \geq 2$, to cells $(i + 1, j)$ or to $(i + 1, j + 1)$ if $j \leq m - 1$. Design an efficient algorithm that finds a path from the bottom row to the top row whose sum of all numbers on the path is the smallest.

4. (20 points) Suppose you are given a set of $n$ integers in the range $0$ to $K$. Design an efficient algorithms that partitions these integers into two subsets which minimizes $|S_1 - S_2|$, where $S_1$ and $S_2$ denote the sums of the elements in each of the two subsets.

5. (20 points) The 3-PARTITION problem is: Given positive integers \(\{a_1, a_2, \ldots, a_n\}\) with $s = a_1 + a_2 + \cdots + a_n$, decides whether it is possible to partition \(\{1, 2, \ldots, n\}\) into three disjoint subsets \(I, J\) and \(K\) such that

\[
\sum_{i \in I} a_i = \sum_{i \in J} a_i = \sum_{i \in K} a_i = \frac{s}{3}.
\]

Give an algorithm for 3-PARTITION problem that runs in time polynomial in $n$ and $s$. 