

# First Homework Assignment

Write the solution to each problem on a single page. The deadline for handing in solutions is September 18.

**Problem 1.** (20 points). Consider two sums,  $X = x_1 + x_2 + \dots + x_n$  and  $Y = y_1 + y_2 + \dots + y_m$ . Give an algorithm that finds indices  $i$  and  $j$  such that swapping  $x_i$  with  $y_j$  makes the two sums equal, that is,  $X - x_i + y_j = Y - y_j + x_i$ , if they exist. Analyze your algorithm. (You can use sorting as a subroutine. The amount of credit depends on the correctness of the analysis and the running time of your algorithm.)

**Problem 2.** (20 = 10 + 10 points). Consider distinct items  $x_1, x_2, \dots, x_n$  with positive weights  $w_1, w_2, \dots, w_n$  such that  $\sum_{i=1}^n w_i = 1.0$ . The *weighted median* is the item  $x_k$  that satisfies

$$\sum_{x_i < x_k} w_i < 0.5 \quad \text{and} \quad \sum_{x_j > x_k} w_j \leq 0.5.$$

- Show how to compute the weighted median of  $n$  items in worst-case time  $O(n \log n)$  using sorting.
- Show how to compute the weighted median in worst-case time  $O(n)$  using a linear-time median algorithm.

**Problem 3.** (20 = 6 + 14 points). A game-board has  $n$  columns, each consisting of a top number, the cost of visiting the column, and a bottom number, the maximum number of columns you are allowed to jump to the right. The top number can be any positive integer, while the bottom number is either 1, 2, or 3. The objective is to travel from the first column off the board, to the right of the  $n$ th column. The cost of a game is the sum of the costs of the visited columns.

Assuming the board is represented in a two-dimensional array,  $B[2, n]$ , the following recursive procedure computes the cost of the cheapest game:

```
int CHEAPEST(int i)
  if i > n then return 0 endif;
  x = B[1, i] + CHEAPEST(i + 1);
  y = B[1, i] + CHEAPEST(i + 2);
  z = B[1, i] + CHEAPEST(i + 3);
  case B[2, i] = 1: return x;
    B[2, i] = 2: return min{x, y};
    B[2, i] = 3: return min{x, y, z}
  endcase.
```

- Analyze the asymptotic running time of the procedure.
- Describe and analyze a more efficient algorithm for finding the cheapest game.

**Problem 4.** (20 = 10 + 10 points). Consider a set of  $n$  intervals  $[a_i, b_i]$  that cover the unit interval, that is,  $[0, 1]$  is contained in the union of the intervals.

- Describe an algorithm that computes a minimum subset of the intervals that also covers  $[0, 1]$ .
- Analyze the running time of your algorithm.

(For question (b) you get credit for the correctness of your analysis but also for the running time of your algorithm. In other words, a fast algorithm earns you more points than a slow algorithm.)

**Problem 5.** (20 = 7 + 7 + 6 points). Let  $A[1..m]$  and  $B[1..n]$  be two strings.

- Modify the dynamic programming algorithm for computing the edit distance between  $A$  and  $B$  for the case in which there are only two allowed operations, insertions and deletions of individual letters.
- A (not necessarily contiguous) *subsequence* of  $A$  is defined by the increasing sequence of its indices,  $1 \leq i_1 < i_2 < \dots < i_k \leq m$ . Use dynamic programming to find the longest common subsequence of  $A$  and  $B$  and analyze its running time.
- What is the relationship between the edit distance defined in (a) and the longest common subsequence computed in (b)?