CPS 230 Homework-1

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

1. **Background Problems:** (20 = 6 + 6 + 8 points)
   (a) Prove by induction that \( \sum_{i=1}^{n} i! = (n+1)! - 1 \).
   (b) For the pair of functions \( f = (\log n)! \) and \( g = \log(n!) \), determine whether \( f = O(g) \)
       and/or \( f = \Omega(g) \) and/or \( f = \Theta(g) \).
   (c) Solve the recurrence \( T(n) = T(n-1) + n^2 \).

2. **Sorting Problem:** (20 = 14 + 6 points)
   Consider sorting a linear array \( A[1...n] \) with BUBBLESORT implemented with a while loop as follows.
   
   \[
   \begin{align*}
   x &= \text{TRUE}; \\
   \text{while } x = \text{TRUE do} & \text{\hspace{2cm}} x = \text{FALSE}; \\
   \text{for } i = 1 \text{ to } n - 1 & \text{ do} \\
   \text{if } A[i] > A[i + 1] & \text{ then} \\
   A[i] & \leftrightarrow A[i + 1]; x = \text{TRUE} \\
   & \text{endif} \\
   & \text{endfor} \\
   & \text{endwhile.}
   \end{align*}
   \]

   (a) Is it true that after \( i \) iterations of the while-loop the largest \( i \) items are in the correct
       last \( i \) positions of the array? Justify your answer.
   (b) What does your answer to Question (a) imply for the running time of the algorithm?

3. **Selection Problem:** (20 = 10 + 10 points)
   For \( n \) distinct elements \( x_1, x_2, \ldots, x_n \) with positive weights \( w_1, w_2, \ldots, w_n \) such that
   \( \sum_{i=1}^{n} w_i = 1 \), the weighted (lower) median is the element \( x_k \) satisfying
   \[
   \sum_{x_i < x_k} w_i < \frac{1}{2} \text{ and } \sum_{x_i > x_k} w_i \leq \frac{1}{2}
   \]

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

Remember to analyze the running time of your algorithms.
4. Dynamic Programming Problem: (20 points)

A game-board consists of \( n \) columns. Each column consists of two numbers, which we refer to as the top number and the bottom number for that column. The top number can be any positive integer, while the bottom number is either 1, 2, or 3.

The object of the game is to travel (by a series of moves) from the first column all the way to the right of the \( n \)-th column (and off the board). The top number of a column is the cost of visiting that column. The bottom number in a column is the maximal number of columns that the player is allowed to jump to the right in the next move. The cost of a game is the sum of the costs of the visited columns until the player moves off the board.

Let the board be represented in a two-dimensional array \( B[n, 2] \). Describe and analyze an efficient algorithm for finding the cheapest game, using dynamic programming.

5. Greedy Algorithm Problem: (20 points)

Let \( x_0, x_1, \ldots, x_n \) be a set of points on the real line. Describe a greedy algorithm that determines the smallest set of unit-length closed intervals whose union covers all the given points in time \( O(n \log n) \).

6. Bonus Problem: (20 = 14 + 6 points are not included in the hundred percent of credit, but can make up the points you lose in all the 7 homework.)

Consider a permutation of \( \{1, 2, \ldots, n\} \) and let \( \pi(k) \) denote the position of item \( k \) in the permutation. Then \( D = \frac{1}{n} \sum_{k=1}^{n} |\pi(k) - k| \) is the average distance that a number travels during sorting.

(a) What is the expected value of \( D \) if the permutation is chosen at random?

(b) What does the answer to (a) say about sorting algorithms where only adjacent items are interchanged?