ASSIGNMENT 2

DUE: Friday, Feb. 9, before midnight. DO NOT COPY. ACKNOWLEDGE YOUR SOURCES.

Please read http://www.student.cs.uwaterloo.ca/~cs341 for general instructions and policies.

1. [Shuffling an array (10 marks)] Given an array of 2n elements $a_1, a_2, ..., a_n, b_1, b_2, ..., b_n$.
   Give an algorithm to shuffle the array in place (without using more than $O(1)$ space) into $a_1, b_1, a_2, b_2, ..., a_n, b_n$. Demonstrate your algorithm is correct and analyze the time complexity of the algorithm.

2. [Finding a peak (10 marks)] Given an array A of integers. An element in A is a peak if it is not smaller than its neighbors. For elements at the two ends of the array, we consider them having only one neighbor. For example, in an array of $\{5, 3, 6, 2, 1\}$, 5 and 6 are peak elements. Give an $O(\log n)$ time algorithm to find a peak element in A. Show the correctness of the algorithm and analyze its time complexity.

3. [Tiling Game (10 marks)] Given an $n$ by $n$ board where $n$ is of form $2^k$, $k \geq 1$. The board has one missing cell (of size 1x1). An L shape tile is a 2x2 square with one cell of size 1x1 missing (so there are 4 kinds of L shaped tiles). See Figure 1. Give an $O(n^2)$ algorithm to completely fill the board (except for the missing cell) with the L shaped tiles. Prove your algorithm is correct by induction, and analyze the time complexity of your algorithm. Hint: You may want to start with the case when the missing cell is one of the 4 corner cells. Then generalize your solution to the case the missing cell can be anywhere.

![Figure 1: Tiling Game](image)
4. **[Uniting the tribes (10 marks)]** Tribes on the planet Alpha are at war. You are sent there to unite them all. There are $n$ tribes. The cost of negotiating to merge two tribes into one bigger tribe is at the cost of the sum of their populations. You can only merge two tribes each time. Give an algorithm to unite (merge) all $n$ tribes into one at minimum cost. Describe your algorithm and prove that your algorithm always achieves the optimal cost. Additionally your algorithm should run in $O(n \log n)$ time. An $O(n^2)$ time algorithm will receive partial credits.

5. **[Minimum Number of Refills (10 marks)]** You are planning to drive along a pre-determined route. At full tank, your car can go $L$ kilometers before it runs out of gas. There are $n + 1$ gas stations $G_0, G_1, ..., G_n$ along the route. For each $G_i$, you are given its distance from the start point of the route, denoted by $d_i$. We assume you start from $G_0$, with full tank, passing $G_i$’s sequentially, and $G_n$ is at your destination. Design and analyze a greedy algorithm so that you go from $G_0$ to $G_n$ with fewest stops for adding gas. (You may assume $d_{i+1} - d_i \leq L$ to ensure that a solution always exists.)

Example: For the input $d_0 = 0, d_1 = 2, d_2 = 7, d_3 = 9, d_4 = 13, d_5 = 15, d_6 = 16, d_7 = 18, d_8 = 25$, and $L = 10$, the optimal solution is to stop at $\{G_2, G_6\}$ or $\{G_3, G_7\}$

Describe your algorithm, prove that your algorithm is correct, i.e. it always computes an optimal solution. Assume that the gas stations (together with the corresponding distances) are given in a sorted array along the route. Your algorithm should run in $O(n)$ time. Give a brief analysis of time complexity.