CS480S15 Midterm

Write your NetID on your exam booklet. **DO NOT write your name.**

Part 1 (40 Points)

1. (a) [4 pts] Sort the following set of strings in lexicographic order: \{ aaabc, aaa, abc, aaabbc \}.
   (b) [6 pts] What comes immediately after [18, 2, 4, 20, 5, 4] in lexicographic order?

2. Consider the following array of integers: [1, 9, 2, 5, 2, 4].
   (a) [6 pts] Give the table of entries for the Fenwick tree of prefix sums for this array.
   (b) [4 pts] Suppose the second element replaced with 7 (so the array becomes [1, 7, 2, 5, 2, 4]), list only the entries in the Fenwick table that have changed and their updated values.

3. [12 pts] Determine the following constants up to 1 significant figure (that is, the number of digits, and the value of the most significant digit should be correct after rounding). Each sub-problem is worth 3 pts.
   (a) Number of possible subsets for a set of 25 elements.
   (b) The number of times \( f \) is called:

\[
\text{int } [] \text{ arr } = \{0, 0, 1, 1, 1\};
\text{do } \{
\quad f ();
\}\text{ while (nextPermutation(arr));}
\]

where nextPermutation behaves as discussed in class.

   (c) Suppose you have 6 elements in a union-find data structure, represented by numbers 1, 2, 3, 4, 5, 6. The following union operations are run: \( 1 \cup 2, 3 \cup 5, 1 \cup 6, 5 \cup 6 \). What is the depth of the tree structure of the set containing 5? **Assume that no path compression is applied.**

   (d) \( S \& \text{-}S \) for \( S = 27 \).

4. [8 pts] You are given a function which takes in a string of length less than 100 characters and returns whether or not it is a valid mathematical expression, handling only digits, \(+, -, *, /, (, )\). You are certain it will return `false` correctly if the input string contains other characters, and certain that it always completes on time even on the longest input. However, you are uncertain that it will be correct otherwise. List test cases to test the correctness of the function.
Part 2 (60 Points)

In each of the following problems, you will be asked to implement a function in either Java or C++. Please use the function prototype specified in each problem. Please note the following.

- Benign compiler errors (e.g., missing semicolons) will be ignored.
- You may omit imports and #includes.
- In C++, you can assume you are using namespace std and in Java, you may assume that all code you provide is already inside a public class Main.
- You may write helper functions wherever necessary.
- You may use any function or data structure in the Java standard library or C++ STL.
- Assume the memory and time limits of the course and assignments.

1. [20 pts] You are given a grid containing obstacles, represented as a 0-indexed 2D boolean array grid with $1 \leq n, m \leq 1000$ rows and columns, respectively. Element grid[i][j] = false if cell i, j is empty, and true if it contains an obstacle. Given a robot which can only move down (i.e., from grid[i][j] to grid[i+1][j]) and to the right (grid[i][j] to grid[i][j+1]) and avoiding obstacles, implement a function that will return the number of possible paths from grid[0][0] to grid[n-1][m-1] modulo 1 million. Assume both grid[0][0] = false and grid[n-1][m-1] = false.

   Java: int countPaths(boolean[][] grid, int n, int m)
   C++: int countPaths(const vector<vector<bool>>& grid, int n, int m)

   Sample input
   countPaths(
      [{false, false, false}, // 0 0 0
       {false, true, false}, // 0 1 0
       {false, false false}], // 0 0 0
      3, 3);

   Sample output
   2

2. [20 pts] Given two lowercase strings a, b, each of length at most 100 characters, implement a function that returns the length of the longest string x such that a permutation of x is a (not necessarily contiguous) subsequence of a and a (possibly different) permutation of x is a subsequence of b (also not necessarily contiguous).

   Java: int longestPerm(String a, String b)
   C++: int longestPerm(const string& a, const string& b)

   Sample input
   longestPerm("aabyuwgef", "yuzmaagef")

   Sample output
   7
3. [20 pts] Given an array of at most 1000000 integers, each in the range [0, 10000], find the longest contiguous subsequence of distinct integers. If multiple solutions exist, return the first occurring.

**Java:** ArrayList<Integer> longestSubseq(ArrayList<Integer> sequence)
**C++:** vector<int> longestSubseq(const vector<int>& sequence)

<table>
<thead>
<tr>
<th>Sample input</th>
<th>Sample output</th>
</tr>
</thead>
<tbody>
<tr>
<td>longestSubseq({1, 3, 5, 2, 5, 8, 9, 0, 1})</td>
<td>2, 5, 8, 9, 0, 1</td>
</tr>
</tbody>
</table>