CS 460, HOMEWORK 5

INSTRUCTOR: HOA VU

Each question is worth 25 points.

When you are asked to design an algorithm, do the following: a) describe the algorithm, b) explain (or more rigorously prove) why it is correct, and c) provide the running time.

DO NOT look up solutions online. Any violation will be reported. I will be happy to provide some hints or ideas if you get really stuck during class or office hours.

[Erickson] denotes the book by Jeff Erickson (available for free online at `http://jeffe.cs.illinois.edu/teaching/algorithms/`).

[DPV] denotes the book by Dasgupta, Papadimitriou, and Vazirani (the required textbook).

For graph problems: If you use a standard graph algorithm that we went over in class, you don't have to rewrite the algorithm. However, you must specify the graph that you construct from a non-graph input (i.e., which are the vertices, edge? Weighted or unweighted? Directed or undirected? If directed, you must specify the edges' directions).

- **Question 1:** Define P, NP, NP-Complete.
- **Question 2:** We want to count how many times a pattern $Y[1 \ldots m]$ occurs as a subsequence in a sequence $X[1 \ldots n]$. For example, X="abdcabc" and Y = "abc" then Y appears 4 times in X. Design a dynamic programming algorithm to solve this problem.

  Hint: Let $T[i, j]$ be the number of times $Y[1 \ldots j]$ occurs as a subsequence of the sequence $X[1 \ldots i]$.
- **Question 3:** In the independent set problem (I.S), we are given a graph with n vertices and m edges. We want to decide if there are k vertices in the graph in which no two of them are connected by an edge. For this question, assume that I.S is NP-Complete.

  A clique is a graph in which there is an edge between every pair of vertices (i.e., a complete graph). Consider the clique problem (Clique): given a graph, decide if there is a clique of size k in that graph. Show that Clique is in NP. Then, show that Clique is NP-Complete by reducing I.S to Clique.
- **Question 4:** Use depth-first-search to solve the following programming question `https://leetcode.com/problems/number-of-islands/`. Note that this is the same as counting connected components where you treat each point in the grid as a vertex. The instruction is similar to programming questions in previous homework.

$\boxed{\text{Question 1}}$ Define P, NP, NP - complete.

Answer:

- P (polynomial) problems: P problems refer to problems where an algorithm would take a polynomial amount of time to solve, or where Big-O is a polynomial (i.e. O(1), O(n), etc). These are problems that would be considered 'easy' to solve. Example: all basic mathematical operations: addition, subtraction, division, multiplication. Shortest Path Algorithms: Dijkstra, Bellman-Ford, Floyd - Warshall,...
- NP (Non-deterministic Polynomial) Problems: This set of problems cannot be solved in polynomial time. However, they can be verified (or certified) in polynomial time. The solutions of the NP class are hard to find since they are being solved by a non-deterministic machine but the solutions are very easy to verify. These are problems that would be considered 'medium' to solve. Example: SAT, graph coloring,...
- NP - Complete Problems: NP-Complete problems are problems at live in both the NP and NP-Hard class. NP-Complete problems are the hard problems in NP. This means that NP-Complete problems can be verified in polynomial time and that any NP problem can be reduced to this problem in polynomial time. If one could solve an NP-complete problem in polynomial time, then one could also solve any NP problem in polynomial time. Example: SAT, 3SAT, Vertex cover, ...

$\boxed{\text{Question 2:}}$ We want to count how many times a pattern Y[1…m] occurs as a subsequence in a sequence X[1…n]. For example, X = "abdcabc" and Y = "abc" then Y appears 4 times in X. Design a dynamic programming algorithm to solve this problem.
Hint: Let T[i,j] be the number of times Y[1…j] occurs as a subsequence of the sequence X[1…i].

Answer: Suppose T[i,j] is the number of times Y[1…j] occurs as a subsequence of the sequence X[1…i]. Then T[i,0] = 1 since there is

always an empty string in a sequence.
When i < j, T[i,j] = 0 since the subsequence is longer than the sequence.
If X[i] == Y[j] then T[i,j] = T[i-1,j-1] + T[i-1,j] or T[i,j] = T[i-1,j]

Pseudocode:

```
for int i = 0 to n+1 {
     for j = 0 to m + 1 {
         if i < j:
            T[i,j] = 0
         else {
            If (X[i] == Y[j]):
                T[i,j] = T[i-1, j-1] + T[i-1,j]
            Else:
                T[i,j] = T[i-1,j]
         }
     }
}
```

Question 3: In the independent set problem (I.S), we are given a graph with n vertices and m edges. We want to decide if there are k vertices in the graph in which no two of them are connected by an edge. For this question, assume that I.S is NP-complete.

A clique is a graph in which there is an edge between every pair of vertices (i.e., a complete graph). Consider the clique problem (Clique): given a graph, decide if there is a clique of size k in that graph. Show that clique is in NP. Then show that clique is NP-Complete by reducing I.S to Clique.

Answer:

1) Show that clique is in NP.

- Certificate: Let the certificate be a set S consisting of nodes in the clique and S is a subgraph of G(V,E)
- Verification: We need to check if there is a clique size k in the

graph. If |S| >= k then checks whether (u,v) in E for every u,v in S. This will take O(n^2) time.

2) Show that clique is NP - complete by reducing I.S to clique.

We consider the Independent Set Problem is in NP-complete.
Every instance of the I.S consisting of the graph G(V,E) and integer k can be converted to the required graph G′(V′,E′) and k′ of the clique problem. Let construct the graph G′ by the following modifications:
V = V′ that means the vertices of graph G are part of the graph G′.
E′ is the complement of the edges E that is the edges not present in the original graph G. Hence, the graph G′ is the complementary graph of G. The running time to compute the graph G′ is O(|V| + |E|).
Let assume that the graph G contains a clique of size k. The presence of clique implies that there are k vertices in G, where each of the vertices is connected by an edge with the remaining vertices. Then these k vertices are not adjacent to each other in G′ and form an I.S of size k.
We assume that the complementary graph G′ has an independent set of vertices of size k′. None of these vertices shares an edge with any other vertices. When we complement the graph to obtain G, these k vertices will share an edge and become adjacent to each other. Then, the graph G will have clique of size k.
We can say there is a clique of size k in graph G if there is an I.S of size k in G′. That means any instance of the clique problem can be reduced to an I.S problem which is NP-complete problem.
Thus, clique problem is NP-complete.

Question 4: Leetcode question: Number of islands. Note that this is the same as counting connected components where you treat each point in the grid as a vertex. The instruction is similar to programming questions in previous homework.

Submitted Code: 0 minutes ago

Language: java

Edit Code

```java
class Solution {
    public int numIslands(char[][] grid) {
        int island = 0;
        for (int i = 0; i < grid.length; i++){
            for(int j = 0; j < grid[0].length; j++){
                if(grid[i][j]== '1'){
                    dfsCall(grid, i, j);
                    island++;
                }
            }
        }
        return island;
    }

    private void dfsCall(char[][] grid, int i, int j){
        if(i>=0 && j>=0 && i<grid.length && j<grid[0].length && grid[i][j] == '1'){
            grid[i][j] = '0';
            dfsCall(grid, i+1, j);
            dfsCall(grid, i-1, j);
            dfsCall(grid, i, j+1);
            dfsCall(grid, i, j-1);
        }
    }
}
```

Back to problem

Submission Detail

49 / 49 test cases passed.

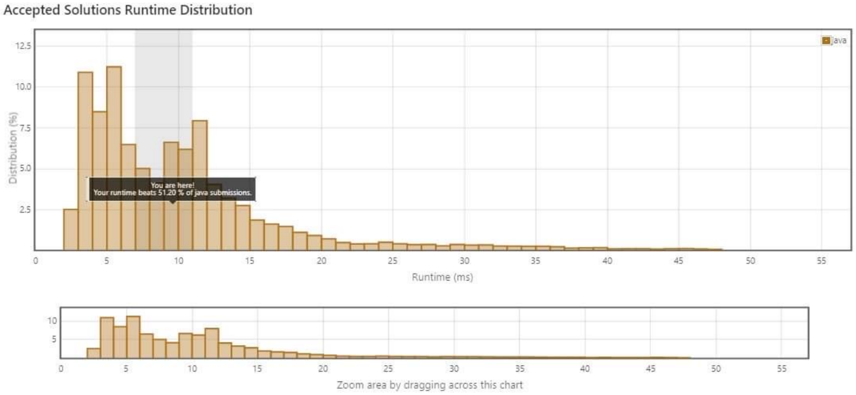
Status: Accepted

Runtime: **9 ms**

Memory Usage: **57.2 MB**

Submitted: **0 minutes ago**

Accepted Solutions Runtime Distribution



Accepted Solutions Memory Distribution