Homework Assignment 3

CSE33101 Intro to Algorithms (Spring 2022)

Due: 2022-06-07 11:59 pm

Question 1, 2, and 3: Handwrite your answer in English, scan it, and submit it to BlackBoard.

Question 4: Type your answer (code) as a source code file and submit it separately to BlackBoard.

Illegible answers or non-compile-able code will not be graded (zero points).

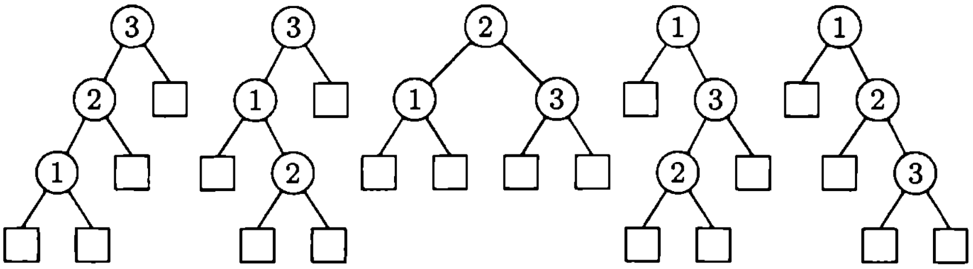
Total 10 points

1. (2 points) Prove that equation $T(n) = 2^n$ is true, given the equation $T(n) = 1 + \sum_{j=0}^{n-1} T(j)$ and the initial condition $T(0) = 1$ of the rod-cutting problem. Use the inductive proof for your solution.

   Hint: $\sum_{k=0}^{n} x^k = \frac{x^{n+1}-1}{x-1}$

2. (3 points) Let's assume that there is a binary search tree having three nodes, and their keys $K_1 < K_2 < K_3$ have respective probabilities 0.2, 0.5, and 0.3 for searching (i.e., we search for the first, second, and third node of the probabilities of 0.2, 0.5, and 0.3). What is the expected number of comparisons required to search for a node?

   Hint: Given three nodes with their keys $K_1 < K_2 < K_3$, there are five possible trees as shown below (by assuming they are equally likely). In the figure, the circles denote each node, and the rectangles denote NIL. For example, when we search for node 1 in the first tree, the number of comparisons required to find it is 3. Similarly, when we search for node 3 in the fourth tree, the number of comparisons is 2.



3. (2 points) Answer the followings.

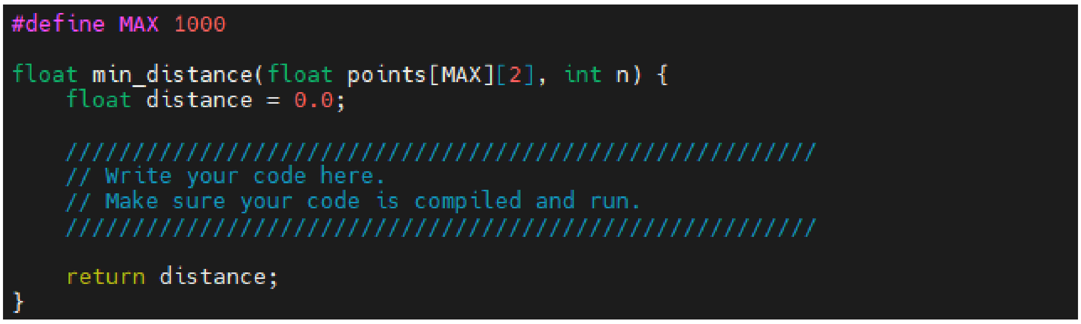
- What is the maximum number of edges in an undirected graph with V vertices and no parallel edges? Parallel edges (also called multiple edges or a multi-edge), are, in an undirected graph, two or more edges that are incident to the same two vertices.

- What is the minimum number of edges in an undirected graph with V vertices, none of which are isolated?

4. (3 points) Let's assume that there is a two-dimensional plane with n number of points represented as (x_n, y_n). We want to connect all the points on the plane in such a way that the total distance of connections (i.e., the summation of distances between all the connected points) is minimized. Write a real code that returns the minimum total distance of all the connected points.

- The distance between two points (x_i, y_i) and (x_j, y_j) is calculated by $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$.
- A point (x_n, y_n) and the distance between two points are real numbers, not integers.
- Your code should be written either in **C or Python**. Your code should be compiled and run. It is all your responsibility to make sure that your code is error-free and has no typos.
- Do not use built-in libraries provided by C and Python, which can directly solve the problem.
- **Upload your code as a separate file** in the same way we did with assignment 2.
- Only the code submitted will be graded (not the code on your computer).
- The function prototype you need to write is given below (C and Python).

i. C



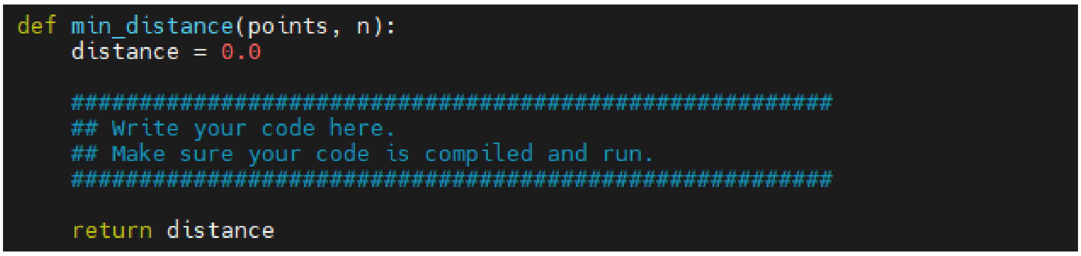
```c
#define MAX 1000

float min_distance(float points[MAX][2], int n) {
    float distance = 0.0;

    ////////////////////////////////////////////////////////////
    // Write your code here.
    // Make sure your code is compiled and run.
    ////////////////////////////////////////////////////////////

    return distance;
}
```

ii. Python



```python
def min_distance(points, n):
    distance = 0.0

    ############################################################
    ## Write your code here.
    ## Make sure your code is compiled and run.
    ############################################################

    return distance
```

iii. The first input parameter "points" are the array of points (x,y), e.g., points[i][0] is the x component of the i-th point, and points[i][1] is the y component of the i-th point.

iv. The second input parameter “n” is the total number of points on the plane.

v. The return value “distance” is the minimum total distance of all the connected points (your solution).