**Midterm**

**Question 1 (12 Points):** Consider the tower of Hanoi problem with 3 Pegs (A, B, C) and 70 Disks. All these disks are placed in Peg A, where the smallest disk is placed on the top and the largest disk is placed on the bottom. We want to move all these disks from A to C.

1. We hire a super human to solve this problem. If the superhuman can perform 10^10 moves per second, what is the total time needed to finish this job? **The minimum amount of moves to complete this problem would be 2^70 – 1, or 1,180,591,620,717,411,303 moves. If we divide this by 10^10 (the moves the superhuman can use) the problem can be solved in 3743 years.**
2. II. Now, consider the tower of Hanoi problem with 4 Pegs (A, B, C, D), and n Disks. All these disks are placed in Peg A. We want to move all these disks from A to D (using B and C as temporary peg). Does this 4-pegs setting still need as many moves as needed by the 3-pegs settings, or the number of moves will be increased or decreased? Justify your explanation? **Since you have one extra peg that has to be used, the total amount of moves would be increased by one.**
3. III. Again, consider the tower of Hanoi problem with 2 Pegs (A, B), and n Disks. All these disks are placed in Peg A. We want to move all these disks to B. What is the minimum number of moves required to solve this problem? **The minimum number of moves would be (2^70) – 1, or 1,180,591,620,717,411,303 moves.**

**Question 2 (10 Points)**: Consider the following recursive code snippet. This code will count the number of vowels in a string. Assume, the string contains only lowercase letters. Answer the following questions:

1) Will this code work for the following inputs: hsdafijbf, ojsbdhfabdj, eaioiia, kjyhgtrwsc? If yes, what will be the output? **The code will not work.**

2) If you think the code will NOT WORK, what is the problem? Fix the problem. **The s.length() – 2 needs to be changed to s.length() – 1.**

**Question 3 (14 Points):** Consider the following recursive code snippet:

public static int myRecursive(String str, char s) { if (str.length() == 0) return 0; else if (str.charAt(0) != s) return myRecursive(str.substring(1), s); else return 1 + myRecursive(str.substring(1), s); }

What value will be returned for the following scenarios? If any of them cannot be handled by the above code snippet, mention ”Invalid Scenario” as answer.

1. myRecursive(“Pennsylvania”, ‘n’) **3**
2. (ii) myRecursive(“Washington”, ‘Y’) **0**
3. (iii) myRecursive(“Z&#WW65 hu77%”, ‘7’) **2**
4. (iv) myRecursive(“Mississauga”, ‘ss’) **Invalid Scenario**
5. (v) myRecursive(“YYY”, ‘Y’) **3**
6. (vi) myRecursive(“Mysterious”, ‘MY’) **Invalid Scenario**
7. (vii) myRecursive(“”, ‘0’). **0**

**Question 4 (10 Points):** Consider a singly linked list of length n.

1. For a given node "M" in this list, in the worst case, what is the number of comparisons needed to find M? **Time is O(n).**
2. Let us assume the list is sorted in ascending order (according to the data value of each node). Then, in the worst case, what is the number of comparisons needed to find M? If we want to add a new node “K” in this list, what is the number of comparisons in the worst case? **The worst case scenario to find M and add K would be O(n) time.**

**Question 5 (10 Points):** Here is a wrong pseudocode which is supposed to determine whether a sequence of parentheses is balanced. Modify the pseudocode to correct it.

**if (stack = empty)**

**print “balanced”**

**else**

**print “unbalanced” and exit**

**Question 6 (12 Points):** Consider the key-set, K=[AGAC, BDDE, WSST, LWMK, IEJK]. A hash function is used to convert these keys into integers. This function assigns a numerical value for each character according to their alphabetical order (A=1, B=2,...., Z=26) and adds each value. For example, AGAC will be converted to 12 (1+7+1+3). Let we transform the key-set K into a set of integers using the hash function mentioned above and want to store these keys using the direct address table technique.

1. What should be minimum size of the table so that it can hold all the values? **27**
2. ii. Lets we add a new key ZSBE and MYZW. What should be minimum size of the table so that it can hold all the values? **29**
3. iii. Will there be any Collison after we add the key in Step II? **No**
4. iv. Will there be any Collison after we add the key WXHM? **No**

**Question 7 (18 Points):** Let there is an undirected graph with 10 nodes with the following edges (x-y means x and y are connected): 1-4, 1-2, 2-3, 2-8, 2-5, 2-7, 3-9, 3-10, 4-3, 5-6, 5-7, 5-8, 8-7. Now,

1. Draw the adjacency matrix representation of this graph.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nodes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 | 0 | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | **1** | 0 | **1** | 0 | **1** | **1** | 0 | 0 |
| 3 | 0 | **1** | 0 | **1** | 0 | 0 | 0 | 0 | **1** | **1** |
| 4 | **1** | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | **1** | 0 | 0 | 0 | **1** | **1** | **1** | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | **1** | 0 | 0 | **1** | 0 | 0 | **1** | 0 | 0 |
| 8 | 0 | **1** | 0 | 0 | **1** | 0 | **1** | 0 | 0 | 0 |
| 9 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | **1** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

1. Say, each cell in this matrix consume 1 byte of memory. If we use a cell to store data, that cell is utilized. Otherwise the cell is wasted. What is the percentage of memory utilization for this graph.

**26/100 = 26%.**

1. We want to determine the in-degree of a specific node from this table and we plan to do it by looping (while/for/do-while etc). How many times the loop should run (at most)? **100 times.**

**Question 8 (14 Points):** Consider a undirected graph (with no self connections) with 𝑛 nodes, where 𝑛 is an odd number and 𝑛 ≥ 4. Verify and justify the following statements:

1. It is possible to create a graph where all these 𝑛 nodes have even degree. **False, the sum of all degrees in an undirected graph must be even. Therefore, since n is odd, the graph cannot have even degree.**
2. II. It is possible to create a graph where all these 𝑛 nodes have odd degree. You must clearly state whether you agree or disagree with these statements and then provide the reasoning. **True, if we have a graph with an even amount of edges, we can delete an edge to make the degree odd. Therefore, we will have a graph with odd degree.**