**Q1 (15 points) Prove that**

1. **worst case of quick sort = n2  using recursion tree**
2. **What is the complexity if the pivot split list is always 1/10 and 9/10? using recursion tree**
3. **Suppose we alternate lucky, unlucky, lucky, unlucky, lucky, ….**

**L(n)= 2U(n/2) + Θ(n)lucky**

**U(n)= L(n –1) + Θ(n)unlucky**

**What is the time complexity in this case.**

**Q2 (5 points)** compute the following recurrence

T(1) =5. T(n) = 8T(n/8) +n for n = 512.

**Q3 (15 points)** Using master method and iteration method to solve

1. T(n) = 4T(n/2) + 2n (4 points)
2. T(n) = 4T(n/2) + n2 (4 points)
3. T(n) = 4T(n/2) + n3 (4 points)

**Q4 (20 points)**

1. Suppose you are choosing between the following three algorithms: **(15 points)**
2. Algorithm A solves problems by dividing them into five sub problems of half the size, recursively solving each sub problem, and then combining the solutions in linear time.
3. Algorithm B solves problems of size n by recursively solving two sub problems of size

n − 1 and then combining the solutions in constant time.

1. Algorithm C solves problems of size n by dividing them into nine sub problems of size n/3, recursively solving each sub problem, and then combining the solutions in O(n2 ) time.

**What are the running times of each of these algorithms (in big-O notation), and which would you choose?**

1. 3-way-Merge Sort: Suppose that instead of dividing in half at each step of Merge Sort, you divide into thirds, sort each third, and finally combine all of them using a three-way merge subroutine. What is the overall asymptotic running time of this algorithm? (Hint: Note that the merge step can still be implemented in O(n) time.)

**(5 points)**

**Q5 (15 points) Get MST using prim and Kruskal [7.5 for each one]**

12

8

5

1

2

9

3

1

8

7

5

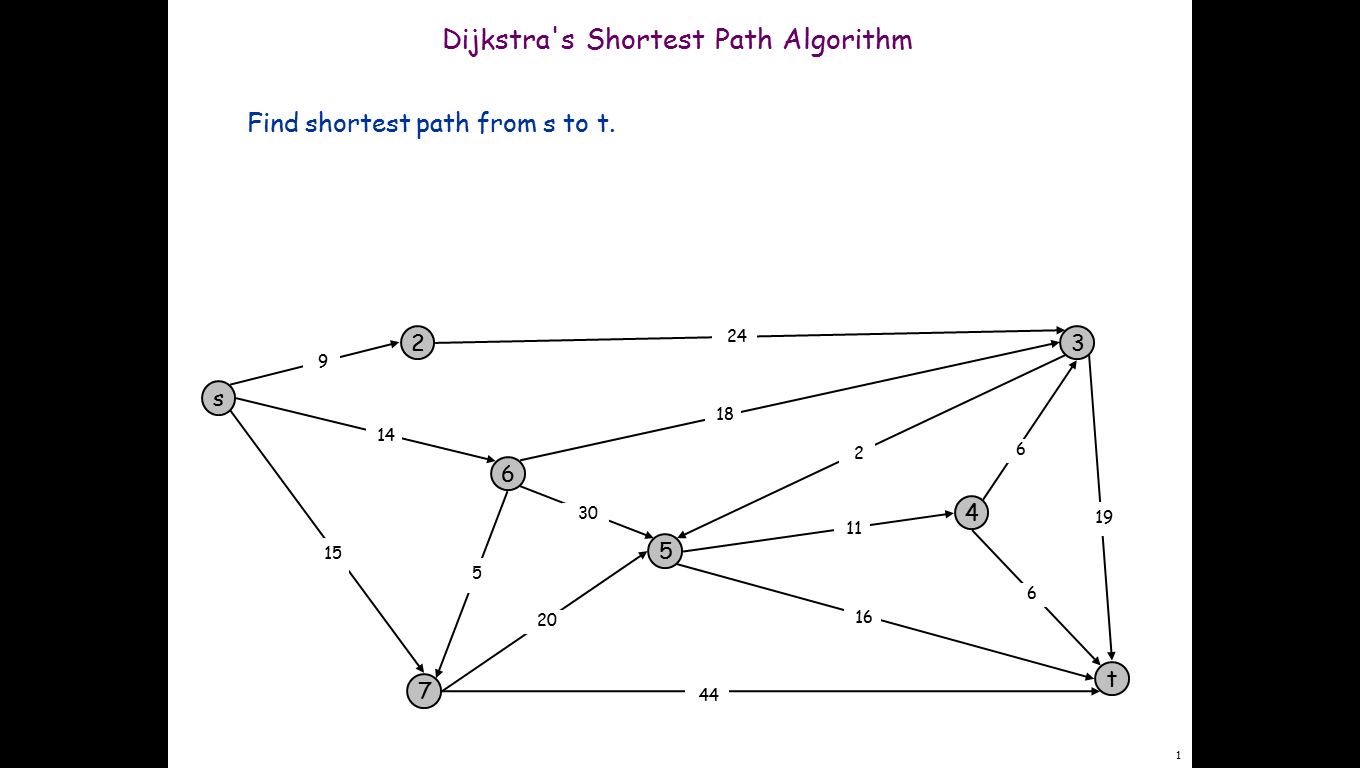
3

9

10

11

**Q6 )10 points) Get shortest paths from node S to each node using Dijkstra algorithm**



**Q7(5 points)** insert the following numbers in a redblack tree (5,1,4,10,9,20,6,7)

, show each step and show the color of each node and its BH.

**Q8 [5 points ]** Construct variable length code using Huffman for the following characters statistics A = 20 , B =30 , C = 15 , D=5, E=8, and F=22 .

**Q9 [10 marks]**

**There is a row of *n* coins whose values are some positive integers c₁, c₂,...,c*n,* not necessarily distinct. The goal is to pick up the maximum amount of money subject to the constraint that no two coins adjacent in the initial row can be picked up. Determine the best selection which giving the maximum amount for the following data 5, 1, 2, 10, 6, 2, using dynamic programming**

**Q1 (5 points)** compute the following recurrence

T(1) =5. T(n) = 5T(n/4) +n for n = 32.

**Q2 (5 points)** Arrange the following functions in increasing order of growth rate

a)n2log(n) b)2n c)22^n d) nlog(n) e)n2

**Q3 (15 points)** Using master method and substitution method to solve

1. T(n) = 4T(n/2) + 2n (4 points)
2. T(n) = 4T(n/2) + n2 (4 points)
3. T(n) = 4T(n/2) + n3 (4 points)

**Q9 [10 marks]** several coins are placed in cells of an n × m board. A robot, located in the upper left cell of the board, needs to collect as many of the coins as possible and bring them to the bottom right cell. On each step, the robot can move either one cell to the right or one cell down from its current location. Design A DP Algorithm to solve this problem, and what is its space and time complexity and use the following counter example to apply your algorithm and what is the maximum number of coins that robot can collect it

Fig 8

|  |  |  |
| --- | --- | --- |
|  |  |  |

***Question (1)* [ 39 points ]**  **Write T for True or F False.**

|  |  |
| --- | --- |
| **#** | **Item** |
| **1** | **For two algorith s A and B, if O(A) = n^100 and O(B) = 10^n, then algorithm A is the better choice** |
| **2** | **To perform a binary search the data set must be sorted.** |
| **3** | **There is a significant difference in the running time f(n) = n and the running time g(n) = lg n.** |
| **4** | **The master method can be used to solve all recurrence equations that have the form T(n) = aT(n/b) + f(n).** |
| **5** | **Printing every node in a balanced tree takes θ(lg n) time** |
| **6** | **In a red-black tree, a black node must have only red children** |
| **7** | **Dynamic programming has the potential to transform exponential-time algorithms to polynomial time** |
| **8** | **In Kruskal's algrorithm for building a minimum spanning tree, we proceed node by node, adding the connecting edges to the tree** |
| **9** | **Huffman encoding uses a dynamic programming strategy to compress data** |
| **10** | **minimum spanning tree has |V| -1 edges** |
| **11** | **Both matrix and lists can be used to represent weighted graphs** |
| **12** | **A matrix is a good choice for representing a dense graph.** |
| **13** | **We do not have to make any key comparisons to find the min or max nodes of a binary search tree.** |

**Question (2) [20 points] Chose the correct answer**

1. **Select the correct asymptotic notation for the function f(n) n^3 + 1000 - 100n^2 - 100n.**

**A. θ(n^3) B.  O(n^3) C.  O(n lg n) D.  Ω(2^n)**

1. **Greedy algorithms make \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_optimal choices leading to \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ optimal solutions.**

**A.  Globally, locally B. Top-down, bottom-up C. Locally, globally D. Bottom-up, top-down**

1. **The basic restructuring operation for red-black trees is**

**A.  Dynamic programming B. Binary sort C. Depth-first search D. Rotation**

1. **Bottom-up dynamic programming works by**

**A. Doubling the recursive calls**

**B. Transforming recursive calls to a loop**

**C. Solving the same subproblem multiple times**

**D. Solving larger subproblems first, followed by smaller subproblems**

1. **What is the name of the technique that stores computed values for future lookup, rather than recomputing them?**

**A recursion tree B Topological sort C. amortize analysis D.Dynamic Programming**

1. **Which of the following is NOT a property of a minimum spanning tree?**

**A It may have one or more cycles. B. It has |V| -1 edges.**

**C. It is not necessarily unique. D. The sum of weights of the edges is minimal.**

1. **Which of the following is true regarding Prim's algorithm for finding a minimum spanning tree?**

**A.  The starting node must be connected to the edge with the least weight.**

**B. The algorithm grows only a single tree from start to finish.**

**C. "Marked" nodes indicate nodes that will not be in the final tree.**

**D. When selecting the next edge, we consider only edges that connect unmarked nodes to the other unmarked nodes.**

1. **What distinguishing feature should cause us to choose dynamic programming rather than a divide-and-conquer approach to a problem?**

**A.  It is an optimization problem. B. The problem can be solved recursively.**

**C. The problem does not have a brute force solution. D. The subproblems for the solution overlap.**

1. **Which one of the following is a true for a red-black tree?**

**A. The root is red.**

**B The root and leaves are black.**

**C All simple paths from a node to the leaves contain the same number of red and black nodes.**

**D A red node can have only one black child.**

1. **The recurrence equation T(n) = T(n-1) + θ(n) represents the**

**A.  Best case running time for Quicksort B. Worst-case running time for Quicksort**

**C. Best-case running time for Heapsort D. Worst-case running time for Heapsort**

**.**

1. **An algorithm that takes constant time is represented by**

**A. Ω(o) B. W(g(n)) C. θ(n) D. θ(1)**

1. **Given the algorithm for Bubblesort below, what is an accurate description of it's running time? BUBBLESORT(A)    for I = 1 to A.length -1       for j = A.length downto I + 1          If A[j] < A[j-1]             exchange A[j] with A[j-1]**

**A.  T(n-1) + θ(n) B.  θ(n^2) C.  2T(n/2) + θ(1) D.  O(n)**

1. **What is the smallest value of n such that an algorithm whose running time is n^2 runs slower than an algorithm whose running time is 3n + 1?**

**A.  2 B. 25 C. 4 D. 15**

**Question (3) [5 point ]**

k-way-Merge Sort. Suppose you are given k sorted arrays, each with n elements, and you want to combine them into a single array of kn elements. Consider the following approach. Using the merge subroutine taught in lecture, you merge the first 2 arrays, then merge the 3rdgiven array with this merged version of the first two arrays, then merge the 4th given array with the merged version of the first three arrays, and so on until you merge in the final (kth) input array.

1. **the** is the running time taken by this successive merging algorithm, as a function of k and n, is**.**

**A. nk B. n2k C. nk2 D. 1**

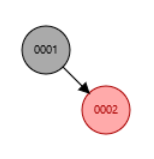
**Question (4)[5 points ]**

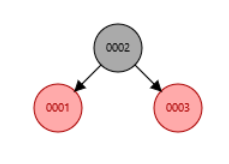
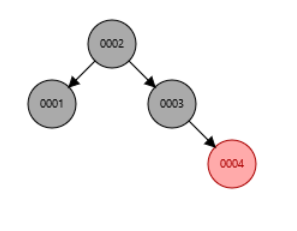
1. **the** amortized analysis for dynamic hash table which start by one size and extend 2 power i I in 0 1 ,2 4 8 ……. By aggregation methods



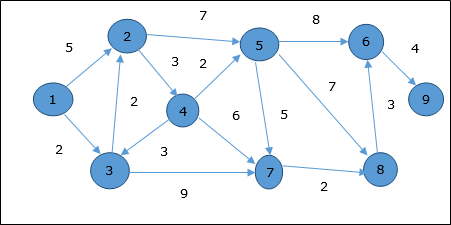
**A. n B. n2 C. nlgn D. 1**

**Question (5) [4 points ]**

1. **inserting these numbers in (1,2,3,4 ) in red-blact tree will give the follwing tree**
2. **b B.**

C. D.

**Question(7) [27 points ]for**

Finding the shortest path for the following graph from node 1 (one) to each nodes using Dijkstra’s algorithm which of the following solution steps are true and which steps are false .

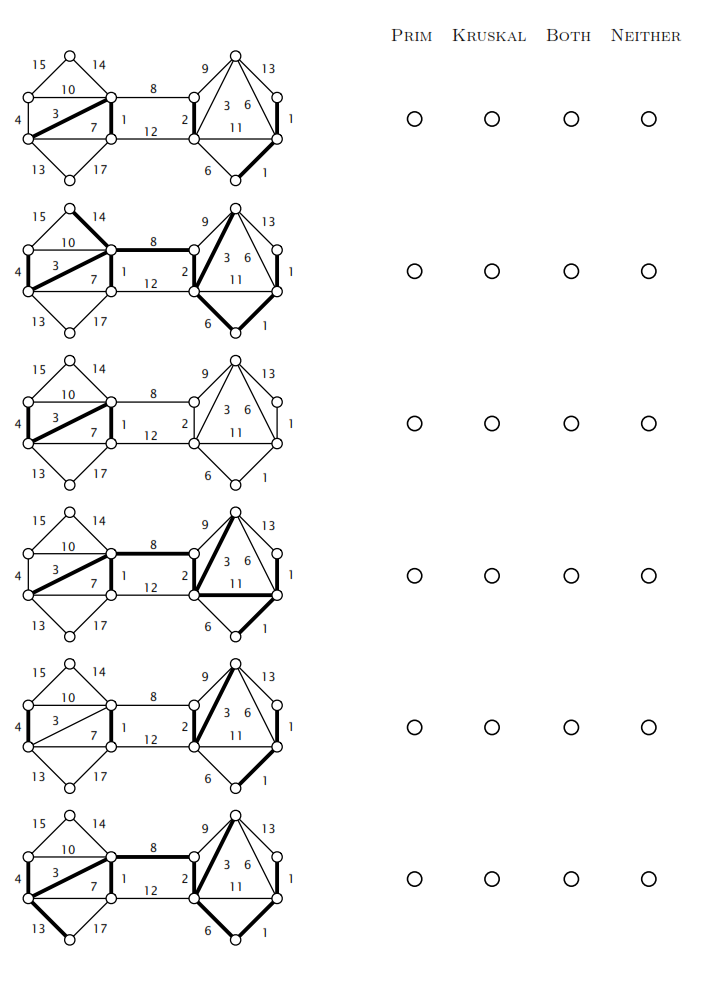
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | parent |  | 3 | 1 | 2 | 4 | 8 | 3 | 7 | 6 |
|  | node | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|  |  | 0 | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ | ∞ |
|  |  |  | 5 | 2 | ∞ | ∞ | 45 | ∞ | ∞ | ∞ |
|  |  |  | 4 |  | ∞ | 46 | ∞ | 11 | ∞ | ∞ |
|  |  |  |  |  | 7 | 11 | 90 | 11 | ∞ | ∞ |
|  |  |  |  |  |  | 9 | ∞ | 11 | 91 | ∞ |
|  |  |  |  |  |  |  | 17 | 11 | 16 | 96 |
|  |  |  |  |  |  |  | 17 |  | 13 | ∞ |
|  |  |  |  |  |  |  | 16 |  |  | ∞ |
|  |  |  |  |  |  |  |  |  |  | 20 |

|  |  |  |
| --- | --- | --- |
|  | *From 1 to 2* | *1-3-2* |
|  | *From 1 to 3* | *1-3* |
|  | *From 1 to 4* | *4-3-2-1* |
|  | *From 1 to 5* | *2-3-1-4-5* |
|  | *From 1 to 6* | *1-3-7-8-6* |
|  | *From 1 to 7* | *1-3-7* |
|  | *From 1 to 8* | *3-1-8* |
|  | *From 1 to 9* | *1-3-7-8-6-9* |

*Paths*

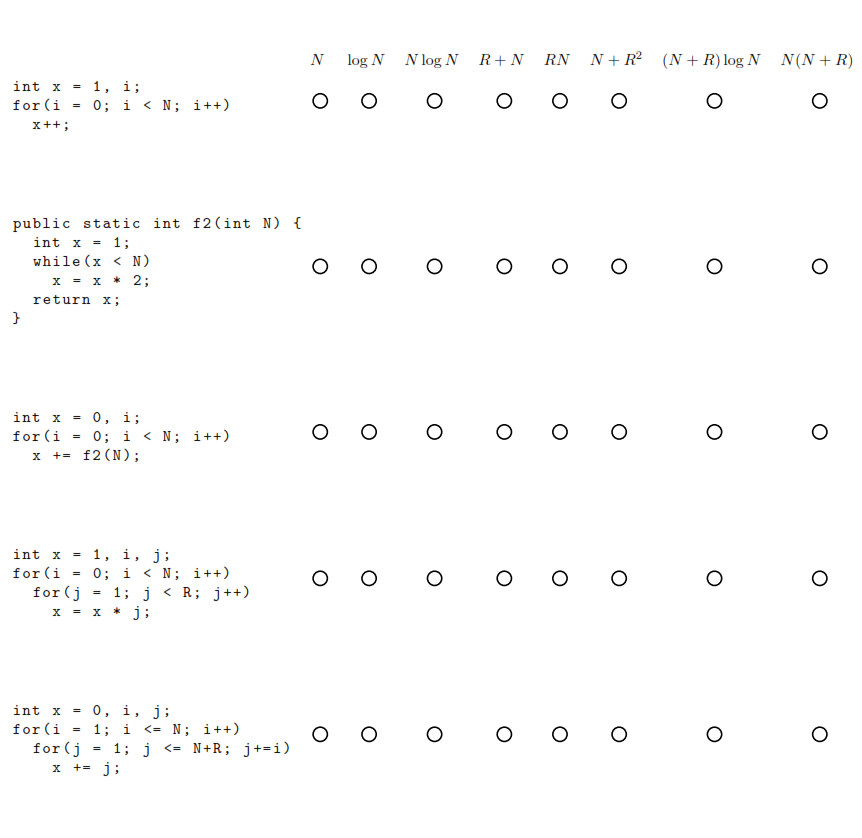
***Question*** (1)**Minimum Spanning Tree Algorithms** (15 points)

Each of the ﬁgures below represents a partial spanning tree. Determine whether it could possibly be obtained from, Prim’s algorithm, Kruskal’s algorithm, both or neither.



***Question*** (**2) Analysis of Algorithms (10 points)**

For each code fragment on the left, check the best matching order of growth of the running time. You may use an answer more than once or not at all. Explain your selection.



***Question*** (3)- five sorting algorithms (10 points)

The column on the left is the original input of strings to be sorted the column on the right are the strings in sorted order; the other columns are the contents at some intermediate step during one of the algorithms listed below. Match up each algorithm by writing its number under the corresponding column. Use each number exactly once, explain your answer.

0 deer bass bass bear bear bull bass

1 clam bull bear bull bull clam bear

2 bear bear bull calf calf bear bull

3 myna crow calf clam clam bass calf

4 tuna deer clam deer deer crow clam

5 slug clam crab dove dove crab crab

6 dove calf crow gnat lynx calf crow

7 moth dove deer lynx moth deer deer

8 lynx hoki dove moth myna lynx dove

9 bull duck duck myna slug moth duck

10 calf crab gnat pony sole dove gnat

11 sole mule hoki seal tuna sole hoki

12 pony moth pony slug gnat pony lion

13 seal lynx seal sole hoki seal lynx

14 gnat gnat myna swan mule gnat moth

15 swan puma swan tuna pony swan mule

16 mule myna mule mule seal mule myna

17 hoki seal sole hoki swan hoki pony

18 duck lion tuna duck bass duck puma

19 crab sole slug crab crab slug seal

20 crow pony lynx crow crow tuna slug

21 bass tuna moth bass duck myna sole

22 lion slug lion lion lion lion swan

23 puma swan puma puma puma puma tuna

---- ---- ---- ---- ---- ---- ----

0 1

(0) Original input (1) Sorted (2) Selection sort (3) Insertion sort

(4) Shellsort (5) Mergesort (6) Quicksort

Q7 You are given a unimodal array of n distinct elements, meaning that its entries are in increasing order up until its maximum element, after which its elements are in decreasing order. Give an algorithm to compute the maximum element that runs in O(logn) time. (5 points) and give an example