Algorithms – FCAI-CU-SWE

1. **What is a hash table?**
   - (A) A structure that maps values to keys
   - (B) A structure that maps keys to values
   - (C) A structure used for storage
   - (D) A structure used to implement stack and queue

2. **If several elements are competing for the same bucket in the hash table, what is it called?**
   - (A) Diffusion
   - (B) Replication
   - (C) Collision
   - (D) Duplication

3. **What is the hash function used in the division method?**
   - (A) h(k)=k/m
   - (B) h(k)=k mod m
   - (C) h(k)=m/k
   - (D) h(k)=m mod k

4. **What can be the value of m in the division method?**
   - (A) Any prime number
   - (B) Any even number
   - (C) $2^p - 1$
   - (D) 2^p

5. **Using division method, in a given hash table of size 157, the key of value 172 be placed at position**
   - (A) 19
   - (B) 72
   - (C) 15
   - (D) 17

6. **What is the table size when the value of p is 7 in multiplication method of creating hash functions?**
   - (A) 14
   - (B) 128
   - (C) 49
   - (D) 127

7. **What is the average retrieval time when n keys hash to the same slot?**
   - (A) Theta (n)
   - (B) Theta (n^2)
   - (C) Theta (n log n)
   - (D) Big-Oh (n^2)

8. **What is Direct Addressing?**
   - (A) Distinct array position for every possible key
   - (B) Fewer array positions then keys
   - (C) Fewer keys than array positions
   - (D) Same array position for all keys

9. **What is the search complexity in direct addressing?**
   - (A) O (n)
   - (B) O (log n)
   - (C) O (n log n)
   - (D) O (1)

10. **What is Hash Function?**
    - (A) A function has allocated memory to keys
    - (B) A function that computes the location of the key in the array
    - (C) A function that creates an array
    - (D) A function that computes the location of the values in the array

11. **Which of the following is not a technique to avoid a collision?**
    - (A) Make the hash function appear random
    - (B) Use uniform hashing
    - (C) Use the chaining method
    - (D) Increasing hash table size

12. **What is the load factor?**
    - (A) Avg. array size
    - (B) Avg. Key Size
    - (C) Avg. chain length
    - (D) Avg. Hash table length

13. **What is Uniform Hashing**
    - (A) Every element has equal probability of hashing into any of the slots
    - (B) A weighted probabilistic method is used to hash elements into the slots
    - (C) Elements has Random probability of hashing into array slots
    - (D) Elements are hashed based on priority

14. in simple chaining, what data structure is appropriate?

Ⓐ Singly Linked list Ⓑ Doubly Linked list Ⓒ Circular Linked list Ⓓ Binary Tree

15. A hash table of length 10 uses open addressing with hash function h(k)=k mod 10, and linear probing. After inserting 6 values into an empty hash table, the table is as shown below. Which one of the following choices gives a possible order in which the key values could have been inserted in the table?

0	
1	
2	42
3	23
4	34
5	52
6	46
7	33
8	
9	

Ⓐ 46,42,34,52,23,33
Ⓑ 34,42,23,52,33,46
Ⓒ 46,34,42,23,52,33
Ⓓ 42,46,33,23,34,52

16. Maximum number of nodes in a binary tree with height k, where root is height 0, is

Ⓐ $2^k - 1$ Ⓑ $2^{k+1} - 1$ Ⓒ $2^{k-1} - 1$ Ⓓ $2^k + 1$

17. Quick sort algorithm is an example of

Ⓐ Greedy Approach Ⓑ Improved binary Search Ⓒ Dynamic Programming Ⓓ Divide and Conquer

18. The minimum number of edges required to create a cyclic graph of n vertices is

Ⓐ n Ⓑ n-1 Ⓒ n+1 Ⓓ 2n

19. Which of the following is example of not in-place algorithm?

Ⓐ Bubble Sort Ⓑ Merge Sort Ⓒ Insertion Sort Ⓓ All of the above

20. Time required to merge two sorted lists of size m and n, is

Ⓐ O (m/n) Ⓑ O (m + n) Ⓒ O (m log n) Ⓓ O (n log m)

21. Which of the following uses memorization?

Ⓐ Greedy Approach Ⓑ Divide and Conquer Ⓒ Dynamic programming Ⓓ None of the above

22. Heap is an example of

Ⓐ Complete Binary Tree Ⓑ Spanning tree Ⓒ Sparse Tree Ⓓ Binary Search Tree

23. Access time of a binary search tree may go worse in terms of time complexity upto

Ⓐ O (n^2) Ⓑ O (n log n) Ⓒ O (n) Ⓓ O (1)

24. What will be the Huffman code for the letters a,b,c,d,e?
if The probability are 1/2 ,1/4 , 1/8, 1/16 ,, 1/32

Ⓐ 0,10,110,1110,1111 Ⓑ 10,011,11,001,010 Ⓒ 10,01,0001,100,1010 Ⓓ 100,110,001,000,010

25. From the following sorting algorithms which algorithm needs the minimum number of swaps

Ⓐ Bubble Sort Ⓑ Quick Sort Ⓒ Merge Sort Ⓓ Selection Sort

26. From the following sorting algorithms which has the lowest worst case complexity?

Ⓐ Bubble Sort Ⓑ Quick Sort Ⓒ Merge Sort Ⓓ Selection Sort

27. What shall be value of x in the following pseudocode?

```
x = 0;
For (t = 1; t< N; t++)
    For( p = 1; p<= t; p++)
        x = x+1
    End For loop;
End For Loop;
```

Ⓐ x=N+1 Ⓑ x=N(N-1)/2 Ⓒ x=N(N+1)/2 Ⓓ $x=N^2$

28. **Match the following pairs**

I.	O(log n)	(M)	Heap sort
II.	O(n)	(N)	DFS
III.	(nlogn)	(O)	Binary search
IV.	$O(n^2)$	(P)	Selecting K^{th} smallest elements

(A) I-P, II-M, III-N, IV-O
(B) I-O, II-P, III-M, IV-N
(C) I-O, II-N, III-M, IV-P
(D) I-O, II-N, III-P, IV-M

29. **Time complexity of merging three sorted lists of sizes m, n and p is**

(A) Θ (m+n+p) (B) Θ (min(m,n,p)) (C) Θ (max(m,n,p)) (D) Θ (m.n.p)

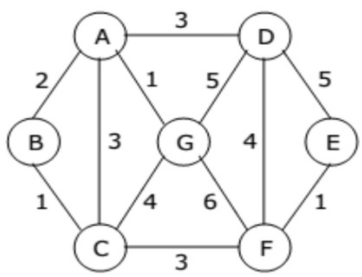
30. **Which of the algorithm design approach is used by Mergesort**

(A) Branch and bound approach
(C) Dynamic approach
(B) Divide and Conquer approach
(D) Greedy approach

31. **Which of the following shortest path algorithm cannot detect -presence of negative weight cycle graph?**

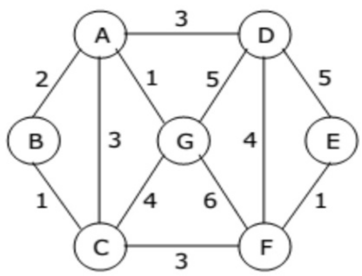
(A) Bellman Ford Algorithm
(C) Dijsktra's Algorithm
(B) Floyd-Warshall Algorithm
(D) None of the above

32. **For the figure below, starting at vertex A, which is a correct order for Prim's minimum spanning tree algorithm to add edges to the minimum spanning tree?**



(A) (A,G) then (G,C) then (C,B) then (C,F) then (F,E) then (E,D)
(B) (A,G) then (A,B) then (B,C) then (A,D) then (C,F) then (F,E)
(C) (A,G) then (B,C) then (E,F) then (A,B) then (C,F) then (D,E)
(D) (A,G) then (A,B) then (A,C) then (A,D) then (A,D) then (C,F)

33. **For the figure below, which is a correct order for Kruskal's minimum spanning tree algorithm to add edges to the minimum spanning tree?**



(A) (A,G) then (G,C) then (C,B) then (C,F) then (F,E) then (E,D)
(B) (A,G) then (A,B) then (B,C) then (A,D) then (C,F) then (F,E)
(C) (A,G) then (B,C) then (E,F) then (A,B) then (C,F) then (D,E)
(D) (A,G) then (A,B) then (A,C) then (A,D) then (A,D) then (C,F)

34. **If G is an directed graph with 20 vertices, how many boolean values will be needed to represent G using an adjacency matrix?**

(A) 20 (B) 40 (C) 200 (D) 400

35. **What is the special property of red-black trees and what root should always be?**

(A) a color which is either red or black and root should always be black color only
(B) height of the tree
(C) pointer to next node
(D) a color which is either green or black

36. **Why do we impose restrictions like . root property is black . every leaf is black . children of red node are black all leaves have same black**

(A) to get logarithm time complexity
(B) to get linear time complexity
(C) to get exponential time complexity
(D) to get constant time complexity

37. **What are the operations that could be performed in O(logn) time complexity by red-black tree?**

- Ⓐ insertion, deletion, finding predecessor, successor
- Ⓑ only finding predecessor, successor
- Ⓒ only insertion
- Ⓓ for sorting

38. **Why Red-black trees are preferred over hash tables though hash tables have constant time complexity?**

- Ⓐ no they are not preferred
- Ⓑ because of resizing issues of hash table and better ordering in redblack trees
- Ⓒ because they can be implemented using trees
- Ⓓ because they are balanced

39. **What is the below pseudo code trying to do? Where pt is a node pointer and root pointer**

```
redblack (Node root, Node pt)
  if (root == NULL)
      return n;
  if (pt.data < root.data)
  {
       root.left = redblack(root.left, pt);
       root.left.parent = root;
  } else if (pt.data > root.data ) {
     root.right = redblack(root.right, pt);
     root.right.parent = root;
  }
  return root
```

- Ⓐ insert a node
- Ⓑ delete a node
- Ⓒ search a node
- Ⓓ counter nodes

40. **Dijkstra's Algorithm is used to solve ____________________ problems**

- Ⓐ All pair shortest path
- Ⓑ Network flow
- Ⓒ Single source shortest path
- Ⓓ Sorting

41. **What is the time complexity of Dijikstra's algorithm?**

- Ⓐ O (N)
- Ⓑ O (N^3)
- Ⓒ O (N^2)
- Ⓓ O (log n)

42. **Dijkstra's Algorithm cannot be applied on ______________________**

- Ⓐ Directed and weighted graphs
- Ⓑ Unweighted graphs
- Ⓒ Graphs having negative weight function
- Ⓓ Undirected and unweighted graphs

43. **How many priority queue operations are involved in Dijkstra's Algorithm?**

- Ⓐ 1
- Ⓑ 2
- Ⓒ 3
- Ⓓ 4

44. **How many times the insert and extract min operations are invoked per vertex?**

- Ⓐ 1
- Ⓑ 2
- Ⓒ 3
- Ⓓ 0

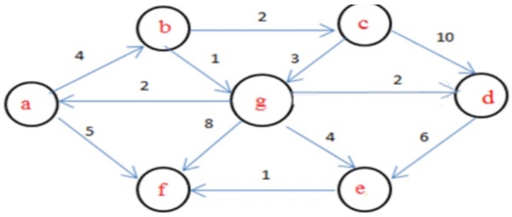
45. **The maximum number of times the decrease key operation performed in Dijkstra's algorithm will be equal to**

Ⓐ Total number of vertices
Ⓑ Total number of edges
Ⓒ Number of vertices – 1
Ⓓ Number of edges – 1

46. **What is running time of Dijkstra's algorithm using Binary min- heap method?**

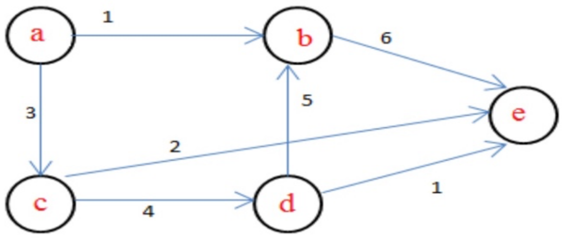
Ⓐ O (V) Ⓑ O (V log V) Ⓒ O (E) Ⓓ O (E log V)

47. **If b is the source vertex,**
what is the minimum cost to reach f vertex?



Ⓐ 8 Ⓑ 9 Ⓒ 4 Ⓓ 6

48. **Identify the shortest path having minimum cost to reach vertex E if A is the source vertex**



Ⓐ a-b-e
Ⓑ a-c-e
Ⓒ a-c-d-e
Ⓓ a-c-d-b-e

49. **Dijkstra's Algorithm is the prime example for**

Ⓐ Greedy algorithm Ⓑ Branch and bound Ⓒ Back tracking Ⓓ Dynamic programming

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

Ⓐ $\theta(n^3)$ Ⓑ $O(n^3)$ Ⓒ $O(n \log n)$ Ⓓ $\Omega(2^n)$

51 **Greedy algorithms make ________ optimal choices leading to ___________ optimal solutions.**

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

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

Ⓐ Dynamic programming Ⓑ Binary Sort Ⓒ Depth-First search Ⓓ Rotation

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

Ⓐ Doubling the recursive calls
Ⓑ Transforming recursive calls to a loop
Ⓒ Solving the same subproblem multiple times
Ⓓ Solving larger subproblems first, followed by smaller subproblems

54. **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

55. **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

56. **You are given infinite coins of denominations v1, v2, v3,.....,vn and a sum S. The coin change problem is to find the minimum number of coins required to get the sum S. This problem can be solved using**

- (A) Greedy algorithm
- (B) Dynamic programming
- (C) Divide and conquer
- (D) Backtracking

57. **Suppose you have coins of denominations 1, 3 and 4. You use a greedy algorithm, in which you choose the largest denomination coin which is not greater than the remaining sum. For which of the following sums, will the algorithm NOT produce an optimal answer?**

- (A) 20
- (B) 12
- (C) 6
- (D) 5

58. **You are given infinite coins of N denominations v1, v2, v3,.....,vn and a sum S. The coin change problem is to find the minimum number of coins required to get the sum S. What is the time complexity of a dynamic programming implementation used to solve the coin change problem?**

- (A) O (N)
- (B) O (S)
- (C) O (N^2)
- (D) O (S*N)

59. **Problems that can be solved in polynomial time are known as?**

- (A) intractable
- (B) tractable
- (C) decision
- (D) complete

60. **__________ is the class of decision problems that can be solved by non-deterministic polynomial algorithms?**

- (A) NP
- (B) P
- (C) Hard
- (D) Complete

61. **To which of the following class does a CNF-satisfiability problem belong?**

- (A) NP Class
- (B) P Class
- (C) NP Complete
- (D) NP hard

62. **Assuming P != NP, which of the following is true?**

- (A) NP-complete = NP
- (B) NP-complete $\cap$P = Φ
- (C) NP-hard = NP
- (D) P = NP-complete

63. **Which of the following is NOT a property of 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.

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

- Ⓐ It is an optimization problem
- Ⓑ The problem can be solved recursively.
- Ⓒ The problem does not have a brute force solution.
- Ⓓ **The subproblems for the solution overlap.**

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

- Ⓐ the root is red
- Ⓑ **the root and leaves are black**
- Ⓒ All simple paths from a node to the leaves contain the same number of red and black nodes.
- Ⓓ A red node can have only one black child.

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

- Ⓐ Best case running time for Quicksort
- Ⓒ Best-case running time for Heapsort
- Ⓑ **Worst-case running time for Quicksort**
- Ⓓ Worst-case running time for Heapsort

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

- Ⓐ $\Omega(0)$
- Ⓑ $\theta(n^2)$
- Ⓒ $\theta(n)$
- Ⓓ **$\theta(1)$**

68. **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]
```

- Ⓐ $T(n-1) + \theta(n)$
- Ⓑ **$\theta(n^2)$**
- Ⓒ $2T(n/2) + \theta(1)$
- Ⓓ $O(n)$

69. **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?**

- Ⓐ 2
- Ⓑ 25
- Ⓒ **4**
- Ⓓ 15

70. *[5 Marks]* **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.**
the is the running time taken by this successive merging algorithm, as a function of k and n, is.

- Ⓐ nk
- Ⓑ n^2k
- Ⓒ **nk^2**
- Ⓓ 1

71. *[5 Marks]* **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**

- Ⓐ n
- Ⓑ n^2
- Ⓒ n log n
- Ⓓ **1**

Let $c_i =$ the cost of the ith insertion

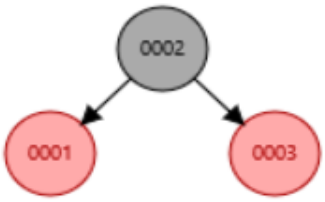
$$= \begin{cases} i & \text{if } i-1 \text{ is an exact power of } 2, \\ 1 & \text{otherwise.} \end{cases}$$

i	1	2	3	4	5	6	7	8	9	10
$size_i$	1	2	4	4	8	8	8	8	16	16
c_i	1	1 1	1 2	1	1 4	1	1	1	1 8	1

72. *[5 Marks]* **inserting these numbers in (1,2,3,4) in red-blact tree will give the follwing tree**

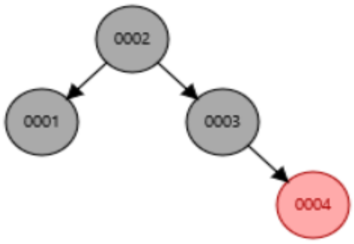
Ⓐ

Ⓒ



Ⓑ

Ⓓ



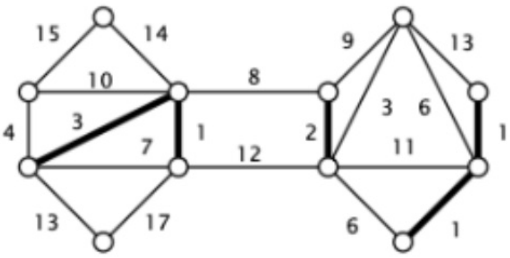
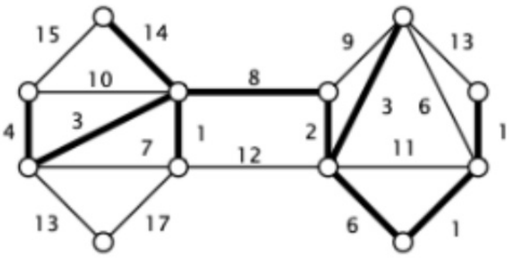
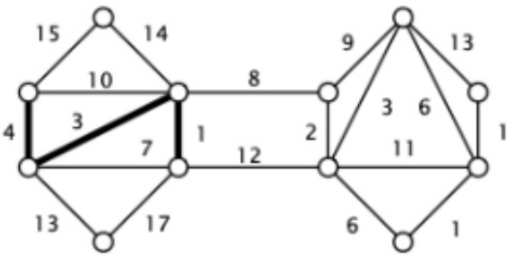
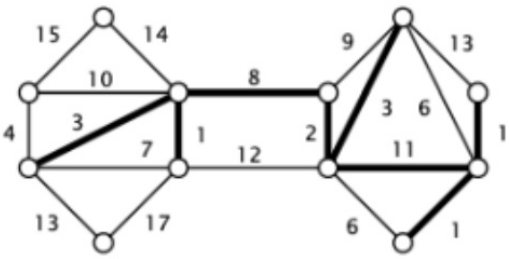
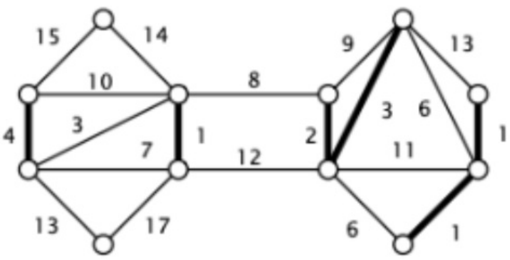
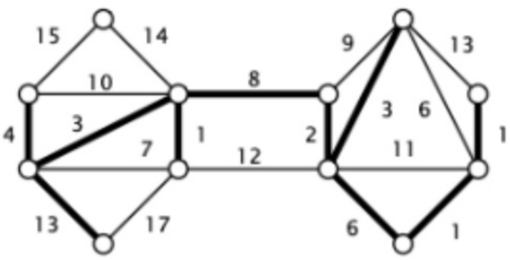
73. *[10 Marks]* **The column on the left is the original input of strings to be sorted the column on the right are the strings in sorted order; the other columns are the contents at some intermediate step during one of the algorithms listed below. Match up each algorithm by writing its number under the corresponding column. Use each number exactly once.**

Answer [Not MCQ]:
- **(0) Original input**
- **(1) Sorted**
- **(2) Selection sort**
- **(3) Insertion sort**
- **(4) Shell sort**
- **(5) Merge sort**
- **(6) Quicksort**
- **(7) Heapsort**

0	deer	bass	bass	bear	bear	bull	tuna	bass
1	clam	bull	bear	bull	bull	clam	swan	bear
2	bear	bear	bull	calf	calf	bear	sole	bull
3	myna	crow	calf	clam	clam	bass	myna	calf
4	tuna	deer	clam	deer	deer	crow	lion	clam
5	slug	clam	crab	dove	dove	crab	slug	crab
6	dove	calf	crow	gnat	lynx	calf	seal	crow
7	moth	dove	deer	lynx	moth	deer	mule	deer
8	lynx	hoki	dove	moth	myna	lynx	lynx	dove
9	bull	duck	duck	myna	slug	moth	crow	duck
10	calf	crab	gnat	pony	sole	dove	clam	gnat
11	sole	mule	hoki	seal	tuna	sole	puma	hoki
12	pony	moth	pony	slug	gnat	pony	pony	lion
13	seal	lynx	seal	sole	hoki	seal	dove	lynx
14	gnat	gnat	myna	swan	mule	gnat	gnat	moth
15	swan	puma	swan	tuna	pony	swan	moth	mule
16	mule	myna	mule	mule	seal	mule	deer	myna
17	hoki	seal	sole	hoki	swan	hoki	hoki	pony
18	duck	lion	tuna	duck	bass	duck	duck	puma
19	crab	sole	slug	crab	crab	slug	crab	seal
20	crow	pony	lynx	crow	crow	tuna	bull	slug
21	bass	tuna	moth	bass	duck	myna	bass	sole
22	lion	slug	lion	lion	lion	lion	calf	swan
23	puma	swan	puma	puma	puma	puma	bear	tuna
	----	----	----	----	----	----	----	----
	0	**4**	**2**	**3**	**5**	**6**	**7**	1

74. *[12 Marks – Very Important]* **Minimum Spanning Tree Algorithms:**

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

	Prim	Kruskal	Both	Neither
	○	○	○	○
	○	○	○	○
	○	○	○	○
	○	○	○	○
	○	○	○	○
	○	○	○	○

75. *[10 Marks]* **Analysis of Algorithms:**
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.

	N	$\log N$	$N \log N$	$R+N$	RN	$N+R^2$	$(N+R)\log N$	$N(N+R)$

`int x = 1, i;`
`for(i = 0; i < N; i++)`
`  x++;`
	●	○	○	○	○	○	○	○

`public static int f2(int N) {`
`  int x = 1;`
`  while(x < N)`
`    x = x * 2;`
`  return x;`
`}`
	○	○	○	○	○	○	○	○

`int x = 0, i;`
`for(i = 0; i < N; i++)`
`  x += f2(N);`
	○	○	○	○	○	○	○	○

`int x = 1, i, j;`
`for(i = 0; i < N; i++)`
`  for(j = 1; j < R; j++)`
`    x = x * j;`
	○	○	○	○	○	○	○	○

`int x = 0, i, j;`
`for(i = 1; i <= N; i++)`
`  for(j = 1; j <= N+R; j+=i)`
`    x += j;`
	○	○	○	○	○	○	○	○

1.	Collisions can be reduced by choosing a hash function randomly in a way that is independent of the keys that are actually to be stored	T
2.	The sum and composition of two polynomials are always polynomials.	T
3.	A non-deterministic algorithm is said to be non-deterministic polynomial if the time-efficiency of its verification stage is polynomial.	T
4.	For two algorithm s A and B, if O(A) = n^{100} and O(B) = 10^n, then algorithm A is the better choice	T
5.	To perform a binary search the data set must be sorted.	T
6.	There is significant difference in the running time f(n) = n and the running time g(n)=log n.	T
7.	The master method can be used to solve all recurrence equations that have the form T(n) = aT(n/b) + f(n).	T and F
8.	Printing every node in a balanced tree takes θ (log n) time	F
9.	In a red-black tree, a black node must have only red children	F
10.	Dynamic programming has the potential to transform exponential-time algorithms to polynomial time	T
11.	In Kruskal's algrorithm for building a minimum spanning tree, we proceed node by node, adding the connecting edges to the tree	F
12.	Huffman encoding uses a dynamic programming strategy to compress data	F
13.	minimum spanning tree has |V| - 1 edges	T
14.	Both matrix and lists can be used to represent weighted graphs	T
15.	A matrix is a good choice for representing a dense graph.	T
16.	We do not have to make any key comparisons to find the min or max nodes of a binary search tree.	T

Khalid Shawki
k.shawki@stud.fci-cu.edu.eg

Preference:

1. https://www.sanfoundry.com/data-structure-questions-answers-hash-tables/
2. https://www.geeksforgeeks.org/data-structure-gq/hash-gq/
3. https://www.sanfoundry.com/hashing-functions-multiple-choice-questions-answers-mcqs/
4. https://www.sanfoundry.com/quicksort-interview-questions-answers/
5. https://www.sanfoundry.com/dijkstras-algorithm-multiple-choice-questions-answers-mcqs/
6. https://www.sanfoundry.com/data-structure-questions-answers-red-black-tree/
7. https://www.sanfoundry.com/prims-algorithm-multiple-choice-questions-answers-mcqs/
8. https://examradar.com/data-structure-graph-mcq-based-online-test-2/
9. https://www.sanfoundry.com/p-np-np-hard-np-complete-complexity-classes-multiple-choice-questions-answers-mcqs/
10. https://www.sanfoundry.com/data-structure-questions-answers-coin-change-problem/
11. https://engineeringinterviewquestions.com/mcqs-on-dynamic-programming/
12. https://testbook.com/objective-questions/mcq-on-dynamic-programming--5eea6a0c39140f30f369e0db
13. https://www.sanfoundry.com/hashing-functions-multiple-choice-questions-answers-mcqs