Solution of Week-0

① Answer - (b)

Explanation - $44 = 32 + 8 + 4$

$= (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$

$\therefore (44)_{10} = (101100)_2$

② Answer - (d)

Explanation - $54 = 3 \times 16 + 6 \times 16^0$

$\Rightarrow (54)_{10} = (36)_{16}$

③ Answer - (c)

④ Answer - (a)

⑤ Answer - (b)

⑥ Answer - (a)

⑦ Answer - (c)

⑧ Answer - (c)

Explanation - According to precedence of operators, $\times$ is evaluated first. $+$ and $-$ have equal priorties.

Hence $1 + 6 - 2 = 5$

⑨ Answer - (d)

Explanation - $(1010111011)_2$

$= (1 \times 2^0) + (1 \times 2^1) + (0 \times 2^2) + (1 \times 2^3) + (1 \times 2^4) + (1 \times 2^5) + (0 \times 2^6) + (1 \times 2^7) + (0 \times 2^8) + (1 \times 2^9)$

$= 699$

$= 512 + 128 + 56 + 3$

$= 1 \times 8^3 + 2 \times 8^2 + 7 \times 8^1 + 3 \times 8^0$

$= (1273)_8$

⑩ Answer - (b)

Week 1 solution

① We know that $n \log n < n^2 < 2^n < n!$

Ans → ⓒ $f_3 < f_1 < f_4 < f_2$

② For ~~first~~ best case input, run time of insertion sort algorithm is $O(n)$.

Ans → ⓐ Linear.

③ reverse sorted array represents the worst case time complexity in insertion sort which is $O(n^2)$

Ans → ⓑ array sorted in reverse order

④ In insertion sort, after m passes through the array, the first m elements are not the m smallest element.

E.g. 4 2 1 0

S-1 2 4 1 0 (after 1 step).
(2 is not smallest in array)

Ans - ⓑ

⑤ Iteration 0 6 4 8 1 3
Iteration 1 4 6 8 1 3
Iteration 2 4 6 8 1 3
Iteration 3 1 4 6 8 3

$(.25 \times 3) = .75$

Ans → ⓒ .75

(6) given array 27, 19, 33, 15, 4

I-1 19, 27, 33 15, 4

I-2 19, 27, 33, 15, 4

I-3 15, 19, 27, 33, 4

ans → (b) 15, 19, 27, 33, 4

(7) $A(n) \leq W(n)$ for each case

$\therefore$ $W(n) = c\,A(n)$ $c > 0$

$\therefore$ $A(n) = O(W(n))$

Ans - (c) $A(n) = O(W(n))$

(8) worst case time complexity of merge sort is $O(n \log n)$.

Ans → (a) $O(n \log n)$

(9) no. of comparison needed to sort two list of size m and n (in the worst case) is $m + n - 1$

Answer → (d) $m + n - 1$

(10) When all the elements are identical

merge sort → $O(n \log n)$

insertion sort → $O(n)$

$\therefore$ insert sort will take least time

Ans → (b) False

Week 2 solution

(1). Master's theorem is used for solving recurrences.

Ans → (a)

(2) The recurrence relation for matrix multiplication using divide and conquer method is

$$T(n) = 8T(n/2) + O(n^2)$$

using Master's theorem $T(n) = O(n^3)$

Ans → (c)

(3) middle = $[(0+9)/2] = 4$.

∴ 4th element is 6.
left most element is 8 & right most element is 0

∴ $[6, 8, 0] \xrightarrow[\text{sorting}]{\text{after}} [0, 6, 8]$
↑ median of three partitioning.

∴ pivot = 6

Ans (c)

(4) For binary search $T(n) = T(n/2) + O(1)$.

∴ $f(n) = 1$, $n^{\log_2 1} = 1$ ∴ $f(n) = n^{\log_2 1}$.
$[a = 1, b = 2]$

∴ case 2 of master's theorem.

Ans → (b)

(5) It follows the pattern of fibonacci series

$n = 1$. # of strings = 2 (0, 1)
$n = 2$ # " " = 3 (00, 01, 10)
$n = 3$ # " " = 5 (000, 001, 010, 100, 101)
so on

∴ $a_n = a_{n-1} + a_{n-2}$

Ans → (b)

(6) 62, 15, 21, 77, 112, 61, 80

77 is not a pivote since 61 is at right of 77 But 61 < 77

112 can be a pivote for same reason.

Ans → (d)

(7) In strassen's matrix multiplication method the second row, third column is

R_3 →

$$\begin{bmatrix} P_5+P_4-P_2+P_6 & P_1+P_2 \\ P_3+P_4 & P_1+P_5-P_3-P_7 \end{bmatrix}$$

$P_3 + P_4$.

Ans → (d)

(8) we can solve recurrence relation of the form

$T(n) = aT(n/b) + f(n)$ where $a \geq 1$, $b \geq 1$

f is assymptotically positive

Ans → (b) False.

(9) Initially (13), 18, 8, 10, 21, 7, 2, 32, 6, 19
Pivot

after partitioning 6, 8, 10, 7, 2, 13, 21, 32, 18, 19

Ans → (d)

(10) ~~result of~~ 1st case of Master's theorem is

$T(n) = O(n^{\log_b a})$.

Ans → (a)

Week 3 solution

① The worst case of Quick sort is when the array is sorted or reverse sorted.

Ans → ⓒ

② Auxiliary space complexity of randomized quick sort — is $O(\log n)$ which is used for storing call stack formed due to recursion.

Ans — ⓒ

③ to sort 1024 elements → $1024 \log 1024$ comparisons

$= 10240$.

to sort 512 elements $= 512 \log 512$ comparisons

$= 4608$

$\therefore$ 10240 comparisons in 200 sec

4608 comparisons in $\frac{200 \times 4608}{10240}$ sec $= 90$ sec

Ans → ⓐ

④ R-Quick sort takes $O(n^2)$ time for identical elements.

R-quick sort is in place.

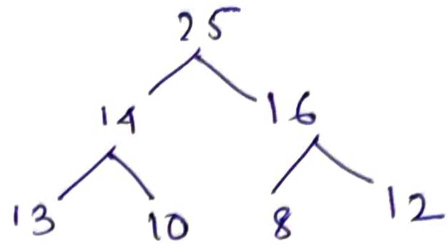
$\therefore$ only statement 2 is correct

Ans → ⓔ

⑤ Time complexity of build heap is $O(n)$

Ans → ⓑ

(6) 25, 14, 16, 13, 10, 8, 12 →



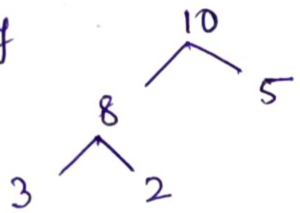
which is a max heap

Ans → (c)

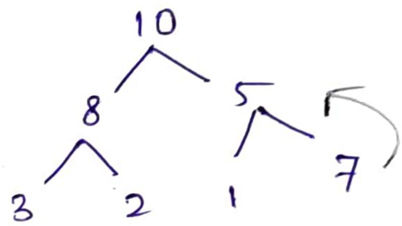
(7) In a max heap containing n elements, the smallest element can be found in O(n).
Since smallest element are present in leaf node so in worst case all the leaf nodes are to be checked.

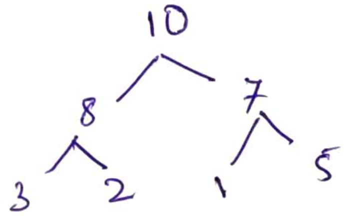
Ans → (a)

(8) initially



$\xrightarrow{1,\,7}$





→ [10, 8, 7, 3, 2, 1, 5]

Ans → (a)

(9) Any decision tree that can sort n elements must have height $\Omega(n \log n)$

Ans → (b) False

(10) parent of a node at index i is in the position $\lfloor i/2 \rfloor$

ans → (d)

Week-4 solution

① Counting sort is a non comparison sort. so number of comparison is 0.

Ans → (a)

② The auxiliary space complexity of counting sort is $O(n+k)$ where k = range of input

Ans → (a)

③ Radix sort is generally implemented as an out of place algorithm, since it needs to create a second, copied array in order to handle the work of sorting.

ans → (b) False

④ Bucket sort is not an inplace sorting algo.

ans → (d)

⑤ Bucket sort is most efficient in the case when the input is uniformly distributed.

ans → (b)

⑥ The run time of SELECT algorithm is optimal for group size 5. & for group size 3 $T(n) > O(n)$ ∴ $T(n) = O(n)$ for group size 5.

Ans → (b)

(7) Counting sort algorithm is efficient when range of data to be sorted is fixed.

Since the range is from 0 to 255 (for ASCII)

~~so~~ so the range is fixed.

Ans → (d)

(8) we can find k th smallest element of an unsorted array in O(n) time in worst case using SELECT method.

Ans → (a)

(9) Radix sort is a non comparison based integer sort.

ans → (a)

(10) For the number in the range from 0 to $n^d - 1$,

~~we~~ ~~so~~ radix sort runs in $\Theta(dn)$ time.

Since we know $T(n,b) = O(bn/\log n)$.

When $r = \log n$ in the original formula

$$T(n,b) = O\left(\frac{b}{r}(n + 2^r)\right)$$

then $\Rightarrow b = d \log n \Rightarrow \underline{T(n) = O(dn)}$

when each b-bit word is broken in b/r equal pieces.

Ans → (b)

Assignment - 5 - Solution

(1) Answer - (c)

Explanation - Using hash function $h(x) = x \bmod 10$.

$h(9679) = 9 = h(1989) = h(4199)$

$h(1471) = h(6171) = 1$

therefore, statement (I) and (II) are correct which match with option (c).

(2) Answer - (b)

Explanation - A hash table is used to implement associated arrays which has a key-value pair, so the has table maps keys to values.

(3) Answer - (c)

Explanation - Given that, $h(x) = (ord(x) - ord(A) + 1) \bmod 10$

$h(K) = ((11-1)+1) \bmod 10 = 1$

$h(R) = (18-1+1) \bmod 10 = 8$

$h(P) = (16-1+1)$ " " $= 6$

$h(C) = (3-1+1)$ " " $= 3$

$h(S) = (19-1+1)$ " " $= 9$

$h(N) = (14-1+1)$ " " $= 4$

$h(Y) = (25-1+1)$ " " $= 5$

$h(T) = (20-1+1)$ " " $= 0$

$h(J) = (10-1+1)$ " " $= 0$

$h(M) = (13-1+1)$ " " $= 3$

only J and M are causing the collision.

(4) Answer - (a)

Explanation - In uniform hashing, the function evenly distributes keys into slots of hash table. Also, each key has an equal probability of being placed into a slot, being independent of the other elements already placed. Therefore, the probility of remaining first 3 slots empty for first insertion (choosing 4 to 100 slots) = 97/100.

As next insertions are independent on previous insertion, the probility for next insertions will also be 97/100. The required probability = $(97/100)^3$.

(5) Answer - (b)

Explanation - In division method for creating hash functions, k keys are mapped into one of m slots by taking the remainder of k divided by m.

(6) Answer - (b)

Explanation - In uniform distribution, the function evenly distributes keys into slots of hash table. For given hash functions, we have calculated hash values for keys 0 to 9 as:

Key	$i^2 \bmod 10$	$i^3 \bmod 10$	$11 \cdot i^2 \bmod 10$	$12 \cdot i \bmod 10$
0	0	0	0	0
1	1	1	1	2
2	4	8	4	4
3	9	7	9	6
4	6	4	6	8
5	5	5	5	0
6	6	6	6	2
7	9	3	9	4
8	4	2	4	6
9	1	9	1	8

(7) Answer - (a)

(8) Answer - (c)

Explanation - In a hash table, if several elements are computing for the same bucket then there will be a a clash among elements. This condition is called collision.

(9) Answer - (c)

Explanation - To get hight 6, we need to put either 1 or 7 at root.

So, count can be written as

$$T(n) = 2\,T(n-1) \quad \text{with} \quad T(1) = 1$$

7

/

[1 ... 6]

1

\

[2, ... 7]

Therefore the count is $2^6 = 64$.

(10) Answer - (a)

Assignment - 6 Solution

(1) Answer - (d)
Explanation - As there are 3 numbers (1 3 2), so total of 6 combinations can be formed using three numbers but since (2,1,3) and (2,3,1) are same. So, in total there are 5 randomized binary search tree that can be formed.

(2) Answer - (a)

(3) Answer - (b)
Explanation - In a random mathematical model, the expected value of number of leaves in a randomized binary search tree is found to be exactly $(n+1)/3$ using probability.

(4) Answer - (a)
Explanation - In an AVL tree, the difference between heights of the two child subtrees of any node is at most one. If height differs by more than one, AVL tree performs rotations to balance the tree.

(5) Answer - (b)

(6) Answer - (c)
Explanation - All the tree data structures given in options are balanced, but B-tree can have more than two children.

(7) Answer - (a)
Explanation - Though both trees are balanced, when there are more insertions and deletions to make the tree balanced, AVL trees should have more rotations, it would be better to use red-black, but if more search is required AVL trees should be used.

(8) Answer- (b)

Explanation- In a balance binary tree the heights of two subtrees of every node never differ by more than 1.

(9) Answer- (c)

Explanation- R B tree is used for Linux Kernel in the form of completely fair schedular process in scheduling algorithm. It is used for faster insertions, retrievals.

(10) Answer- (d)

Assignment - 7 Solution

(1) Answer - (d)

Explanation- In order to search a key or integer in the Van - Emde - Boas data structure, the operation can be performed on an associative array. Hence, the time complexity for a searching a key or integer in Van Emde Boas data structure is $O(\log(\log M))$.

2. Answer- (d)

Explanation- The Van - Emde - Boas data structure is also propularly known as Van Emde Boas priority Queue. This data structure implements the array associatively for the given integer keys. It is a non-binary type of tree.

3. Answer - (c)

Explanation- The van-Emde-Boas data structure implements an abstract data type called associative array for the given integer keys.

4. Answer- (c)

(5) Answer- (d)

(6) Answer- (d)

Explanation- Consider a sequence of n insertions. The worst case time to execute one insertion is $\theta(n)$. Therefore, the worst-case time for n insertions is $n\theta(n) = \theta(n^2)$

(7) Answer- (b)

(8) Answer- (d)

(9) Answer- (c)

(10) Answer- (b)

Assignment - 8 Solution

① Answer - (c)
Explanation - The longest common subsequence is 'BCBA'. So, the length is 4.

② Answer - (d)
Explanation - The space complexity of the above dynamic programming implementation of the longest common subsequence is $O(mn)$.

(3) Answer - (b)
Explanation - Row number of the matrix represents the tail, while column number represents the head of the edge.

(4) Answer - (b)

(5) Answer - (b)

(6) Answer - (c)
Explanation - The time complexity of the brute force algorithm used to find the longest common subsequence is $O(2^n)$.

(7) Answer - (b)
Explanation - Only undirected graphs produce symmetric adjacency matrices.

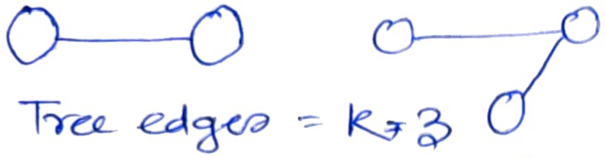
(8) Answer - (d)
Explanation - A graph can have many spanning trees. Each spanning tree of a graph G is a subgraph of the graph G, and spanning trees include every vertex of the graph. Spanning trees are always acyclic.

(9) Answer - (c)
Explanation - Total number of values in the matrix is $4 \cdot 4 = 16$, out of which 6 entries are non-zero.

(10) Answer - (a)
Explanation - All adjacency matrices are square matrices.

Introduction to algorithm and analysis
Assignment 9 solution

(1) Answer — (d)
Explanation — Tree edges are the edges that are part of DFS. If there are x-trees edges in a tree. then, x+1 vertices in the tree.



Tree edges = K=3
vertices n = 5   Connected Components = 2

(2) Answer — b.
Explanation — (a) There can be more than one paths with same weight. Consider a path with one-edges of weight 5 and another path with two edges of weights 2 and 3. both the path have same weights.
(b) Dijkstra and Bellman Ford Algo. both works fine for a graph with all +ve weights. but they are different algo. and may pick differt edges for shortest paths.

(3) Answer — c

(4) Answer — c

(5) Answer — d
Explanation — Dijkstra's algo. single source shortest path is not guranteed to works for graphs with -ve weight edges, but it works for the given graph. Letus see.
Letus run the 1st pass
b 1
b is minimum, so shortest distance to b is 1
After 1st pass, distances are
c 3, e -2
e is minimum, so shortest distance to e is -2

After 2nd pass, distances are
c 3, f 0
f is minimum, so shortest distance to f is 0

After 3rd pass distance are
c 3, g 3.
Both are same, let us take g. so shortest distance to g is 3.

After 4th pass distances are
c 3, h=5,
C is minimum. so shortest distance to c is 3.

After 5th pass, distances are
h -2
h is minimum. So shortest distances to h is -2

(6) Answer - a
Explanation - the shortest path remains same. It is like if we change unit of distance from meter to kilometer the shortest path don't change

(7) Answer - c

(8) Answer - d
Explanation - (i) abeghf (visit a, explore all adjacents through b, and so on). In this b is adjacent e is picked 1st
(iii) abfhge (visit a, explore all adjacents though b, and so on)
In this b's adjacents f is picked 1st.
(iv) afghbe (visit a, explore all adjacents through f and so on)
In this f's adjacent g is picked 1st.
ii) ab·fehg can't be answer. as e is visited after f.

(9) Answer - a

(10) Answer - b.

Introduction to algorithm and analysis
Assignment 10 solution

(1) Answer- c
Explanation:- when the total weight of the graph sum upto a (-ve) number then the graph is said to have a negetive weight cycle.

(2) Answer— b.

(3) Answer— b.
Explanation- There are 4 edges s-a, a-b, b-t and s-t of weights 1, 1, 1 and 4 respectively. The shortest path s to t is s-a, a-b, b-t. If we increase weight of every edges by 1, the shortest path changes to s-t.

(4) Answer— b.

(5) Answer- c

(6) Answer— c
Explanation- Time Complexity of Bellman Ford algo is $\Theta(VE)$ where V, E are numbers of vertices and edges respectively. For complete graph $V = n$, $E = \Theta(n^2)$
So. overall time complexity is - $\Theta(n^3)$

(7) Answer— b.
transitive closure of Graph is

$$t_{ij}^{(k)} = t_{ij}^{(k-1)} \vee \left(t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)}\right)$$

(8) Answer- a
By Applying Floyd warshall algo we get following matrix

$$D^{(0)} = \begin{bmatrix} 0 & 8 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & \infty & 0 & \infty \\ \infty & 2 & 9 & 0 \end{bmatrix} \quad \text{Now, } D^{(1)} = \begin{bmatrix} 0 & 8 & \infty & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 9 & 0 \end{bmatrix}$$

$$D^{(2)} = \begin{bmatrix} 0 & 8 & 9 & 1 \\ \infty & 0 & 1 & \infty \\ 4 & 12 & 0 & 5 \\ \infty & 2 & 3 & 0 \end{bmatrix} \qquad D^{(3)} = \begin{bmatrix} 0 & 8 & 9 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 12 & 0 & 5 \\ \cancel{\infty}\,7 & 2 & 3 & 0 \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & 3 & 4 & 1 \\ 5 & 0 & 1 & 6 \\ 4 & 7 & 0 & 5 \\ 7 & 2 & 3 & 0 \end{bmatrix}$$

which represents the shortest path between every pair of vertices.

(9) Answer – a

Explanation – Recurrence relation of Floyd-Warshall algo. is

$$C_{ij}^{(k)} = \min_k \left\{ C_{ij}^{(k-1)},\ C_{ik}^{(k-1)} + C_{kj}^{(k-1)} \right\}$$

(10) Answer – a

Explanation – Worst case time complexity of Johnson's algo. is – $O(|V|^2 \log|V| + |V||E|)$

Introduction to algorithm and analysis
Assignment 11 Solution

(1) Answer – a
Explanation - Vertex with no incoming edges is called as a source.

(2) Answer – a
Findset operation returns the representative element of a set S_x containing x.

(3) Answer – d

(4) Answer – a

(5) Answer – d
Explanation – Path compression algo is performed during find operation and is independent of strategy used to perform unions.

(6) Answer – c (by defination it follows)

(7) Answer – b.
A simple acyclic path between source and sink which pass through only +ve weighted edges is called augmenting path.

(8) Answer – c

Answer- a

(9) For skew symmetry.

$$f(u,v) = (f_1+f_2)(u,v) = f_1(u,v) + f_2(u,v) = -f_1(v,u)$$

$$-f_2(v,u) = -(f_1+f_2)(v,u) = -f(v,u)$$

Flow conservation —

Let, $u \in V \setminus \{s,t\}$

$$\sum_{v \in V} f(u,v) = \sum_{v \in V} f_1(u,v) + f_2(u,v) = 0.$$

Capacity constraints —

Let, $V = \{s,t\}$ $E = \{(s,t)\}$

$c(s,t) = 1$ Let, $f_1(s,t) = 1$, $f_2(s,t) = 1$

f_1, f_2 obey capacity constraints.

but $(f_1+f_2)(u,v) = 2$ which violates capacity constraints.

(10) Answer- b.

Introduction to algorithm and analysis
Assignment 12 solution

(1) Answer — C
Explanation — Prim's algo is graddy algo. All other's are dynamic problem.

(2) Answer — C
Explanation - In divide and Conquer, the problem is divided into smaller non overlapping subproblems and optimal solution for each of the subproblem is found.

(3) Answer — b.

(4) Answer - b.

(5) Answer - e.

(6) Answer - d
Explanation - (I) is true because cycle detection can be done in polynomial time using DFS.
(II) is true because P is subset of NP.
(III) is true because NP complete is also NP and NP means non deterministic ~~path~~ polynomial Solution exists.

(7) a

(8) Answer - d

(9) Answer - b.

(10) Answer - b.

Explanation - (a) incorrect because R is not NP. A NP- complete problem has to be in both NP and NP-hard. (b) correct because a NP complete problem S is polynomial time educable to R. (c) incorrect because Q is not NP. (d) incorrect because there is no NP complete problem that is polynomial time Turing reduciable to Q.