Indian Institute of Technology Patna

Date: 27 November 2022

Endsem: CS204: Algorithms

Duration: 3 hours Full Marks 60

Instruction: For Q1, proper option should be written on the table 2 provided in the last page. For Q2, you are supposed write the answer in table 3 in the last page.. You are not allowed to write/mark anything in question paper from page 1 - page 6 except your personal details at page 1. If you have any assumption for any question, you may write along with question number in page 6 blank space. Evaluation of Q1-2 will be based only on the tables provided in last page. Q3-Q5 should be answered within given space just after those questions. No doubts would be entertained. Write suitable assumptions, if necessary.

Name (in Capital): ____________________

Roll No: ____________________

Signature: ____________________

Phone Number: ____________________

Q(1) Write the correct option in below table 2. Each question carries 2 marks. Wrong choice will be awarded −0.5. 0 mark will be awarded if any mark is found in page 1-6 and clarity is not maintained. [10X2=20]

(a) Let G = (V, E) be a connected undirected edge-weighted graph. The weights of the edges in E are positive and distinct. Consider the following statements:
I. Minimum Spanning Tree of G is always unique.
II. Shortest path between any two vertices of G is always unique.

Which of the above statements is/are necessarily true?

i. I only
ii. II only
iii. Both I and II
iv. None

(b) Given an array of 7 numbers. We wish to apply Radix sort only on the 1's place digit (least significant position). What would be the sum of arr[1] and arr[5] (assume the array index started from 0) after the Radix sort at position 1 ?
Initial array is arr[] = {35, 915, 64, 128, 11, 311, 610}

i. 1226
ii. 46
iii. 926
iv. 346

(c) Given a sorted array, what can be the minimum worst-case time complexity to find the ceiling of a number x in a given array? The ceiling of an element x is the smallest element present in the array, greater than or equal to x. The ceiling is not present if x is greater than the maximum element present in the array. For eg., if the given array is 12, 67, 90, 100, 300, 399 and x = 95, then output should be 100.

i. O(n)

ii. O($\sqrt{n}$

iii. O(log(logn))

iv. O(logn)

(d) What is the time complexity of the following function?

```
int fun(int n){
  int cnt = 0;
  for (int i = n; i > 0; i /= 2)
     for (int j = 0; j < i; j++)
        cnt += 1;
  return cnt;
}
```

i. O(nlogn)

ii. O(n)

iii. O($n^{\frac{3}{2}}$)

iv. O(n^2)

(e) Consider the following recursive C function that takes two arguments. Choose the correct answer when this function is called for n=513 and r=2.

```
unsigned int foo(unsigned int n, unsigned int r)
   {
      if(n>0) return((n%r)+foo(n/r,r));
      else return 0;
    }
```

i. 9

ii. 8

iii. 2

iv. 5

(f) Consider the directed graph shown in the Figure 1. There are multiple shortest paths between vertex S and vertex T. Which one will be reported by Dijkstra's shortest path algorithm? Assume that, in any iteration, the shortest path to a vertex v is updated only when a strictly shorter path to v is discovered

i. SDT

ii. SBDT

iii. SACDT

iv. SACET

(g) let's say you have an efficient algorithm to find out Hamiltonian path. You want to develop an algorithm of finding Hamiltonian cycle using the algorithm of Hamiltonian path. You came up with an algorithm, where you shall execute a loop for each edge in the graph and at each time you remove an edge from graph and check if there is still a Hamiltonian path in the remaining graph. Point out which of the following statement is false.

i. If original graph contains a Hamiltonian cycle then after removing any edge, you will always get a Hamiltonian path

ii. If you get always a Hamiltonian path after removing an edge that means original graph surely contains a Hamiltonian cycle.

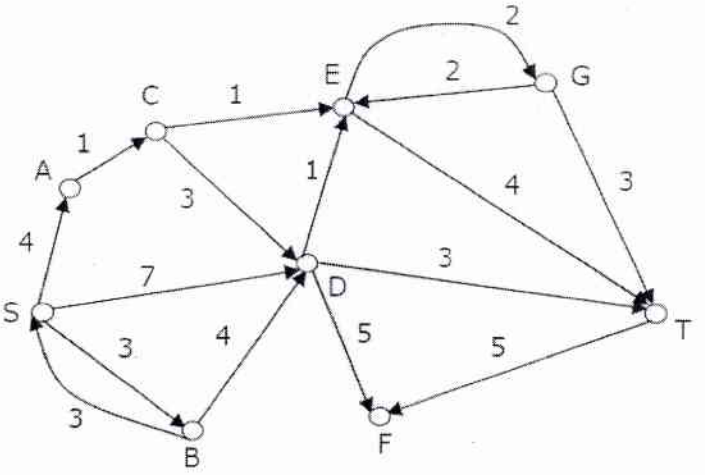


Figure 1: Shortest path

   iii. If the original graph does not contain any Hamiltonian path then after removing an edge, you will never get any Hamiltonian path
   iv. If the original graph does not contain a Hamiltonian cycle, then after removing an edge at a time, you might get Hamiltonian path

(h) Which of the following statements are TRUE? (1) The problem of determining whether there exists a cycle in an undirected graph is in P. (2) The problem of determining whether there exists a cycle in an undirected graph is in NP. (3) If a problem A is NP-Complete, there exists a non-deterministic polynomial time algorithm to solve A.

   i. 1, 2 and 3
   ii. 1 and 3
   iii. 1 and 2
   iv. only 1

(i) Let us say that problem X and problem Y belong to class NP. It is proven that X $\leq_p$ Circuit_SAT and Circuit_SAT $\leq_p$ Y. Then which of the following statement is true?

   i. X and Y are in class NPC
   ii. X is in NPC but can't comment about Y
   iii. Y is in NPC but can't comment about X
   iv. X and Y none of them are in NPC

(j) Regarding Max-Flow which of the following statement is false?

   i. Max-flow of a graph/network is always equal to min-cut of that graph
   ii. During flow augmentation distance of a node from source node monotonically increases.
   iii. As per the flow conservation property amount of input flow to any node (other than source and sink) is always equal to amount of outflow from that node.
   iv. At the time you attained max-flow, at least there exist one path in residual network from source to sink

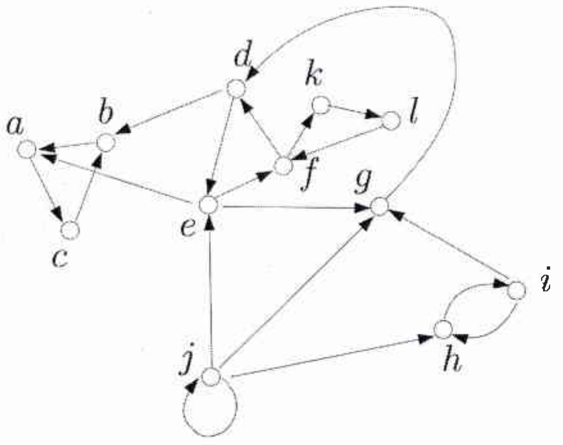


Figure 2: Strongly Connected Component

Q(2) Fill in the blanks/Find answers for following questions. Write your answer in table 3. You are not allowed to write any thing in page 1 - page 6. [15X2=30]

(a) Solve the following recurrence relation T(n) = T(n/4) + T(n/2) + n^2. T(n)=θ (______)

(b) Find the number of interchanges (one interchange is swap of positions between two numbers) required in following array to convert it to a max-heap (70, 23, 60, 19, 13, 16, 1, 4, 8, 12, 7, 10, 85).

(c) Best case time complexity for insertion sort is θ (______) and quick sort is θ (______).

(d) Please fill in the blanks to complete counting sort algorithm

```
Counting-Sort(A,B,k)
     for i: 1 to k
         do C[i]=0
     for j: 1 to length(A)
         do C[A[j]]=C[A[j]] + 1
     for i: 2 to k
         do C[i]= C[i] + C[i-1]
     for j: length(A) downto 1
         do  _________ = A[j]
             C[A[j]] = C[A[j]] -1
```

(e) Find the graph of Figure 2 and find out all strongly connected components.

(f) 25 distinct elements to be sorted using quick sort where pivot is chosen randomly. For the first round of quick sort what is the possibility that chosen pivot will be placed at worst possible location? (rounded off to 2 decimal places)

(g) Find the flow network of Figure 3. Find out the maximum flow and point out one corresponding minimum cut (write set of edge that crosses the cut).

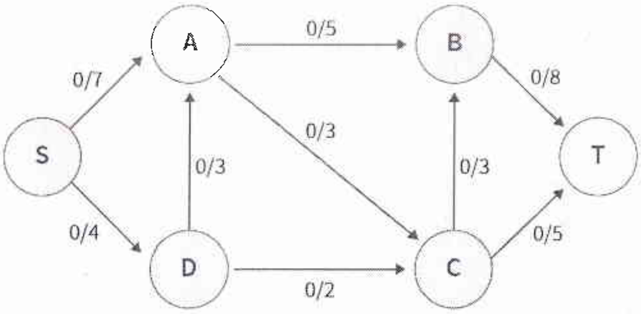


Figure 3: Max-Flow

NIL	3	4	5	1
4	NIL	4	2	1
4	3	NIL	2	1
4	3	4	NIL	1
4	3	4	5	NIL

Table 1: Path Matrix

(h) Let's say there are 4 matrices P, Q, R and S of dimension 13X12, 12X30, 30X15 and 15X18 respectively. What would be the minimum number of scalar multiplication required to find PQRS?

(i) During Floyd-Warshall shortest path algorithm execution, you got the path matrix(indexed from 1 to 5 instead of 0 to 4) as shown in table 1. write down the shortest path between node 1 and node 2.

(j) Approx-Vertex-Cover (G)

```
C= $\phi$
E'=G.E
while E'!= $\phi$
    Find a random edge (u,v) from E'
    C= C $\cup$ {u,v}
    Remove from E' every edge either incident on u or v.
return C
```

Approximation ratio of above vertex-cover algorithm is ______

(k) Suppose we have an array of 12 integers: {13, 19, 9, 5, 12, 8, 7, 4, 21, 2, 6, 11}. What is the output by the end of the first iteration of the Quick sort algorithm using the last element as pivot element?

(l) Let's say in a passage, there are 5 alphabets a, b, c, d, e where exact count of those alphabets are 25, 20, 10, 20, 50 respectively. If you go for Huffman coding, then what would be the cost (in bits) of presenting this passage?

(m) Johnson's Algorithm works fine with negative edge weights. The first step of the algorithm is called Reweighing which uses the Bellman Ford algorithm. This generates a shortest distance of each vertex from a new source vertex and new non-negative weights for each edges. Given the following edge list, write down the

i. Shortest distance from new source vertex to each vertex [2]

ii. The new non-negative edge weights in the prescribed format. [2]

Edges and their weights
a→ b : -2, c→ a : 4, b→ c : -1, c→ x : 2, c→ y : -3, z→ x : 1, z→ y : -4

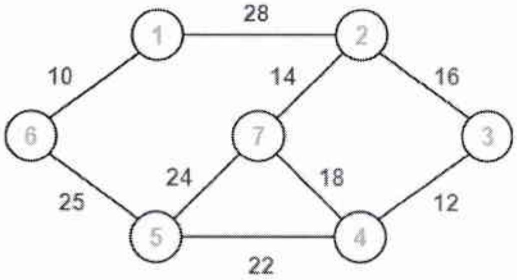
(n) Fill in the blank to complete Bellman-Ford algorithm

```
Bellman_Ford(G,W,s)
    Initialize-single-source(G,s)
    for i: 1 to _________
         for each edge (u,v) from G.E
             Relax(u,v,W)
    for each edge (u,v) from G.E
         if v.d > u.d + w(u,v)
             return False
    return True
```

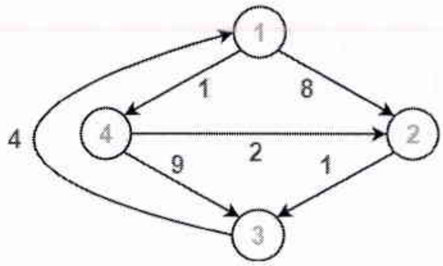
Q(3) Show the red-black trees after successively inserting the keys 41, 38, 31, 12, 19, 8 in an initially empty red-black tree. **Show the structure of only final tree.**

Note: A "Black" node should be represented by inserting few 'b' characters in circle, while "Red" nodes can be represented by inserting few 'r' characters in circle. [3]

Q(4) Find the MCST when Prim's algorithm is used with input graph shown in below Figure and source node is 1. [3]



Q(5) Find the graph of below Figure. Use Floyd warshall algorithm to determine D_1 and D_2. Note: D_i is the matrix where i can be used as intermediate node. [4]



Name (in Capital): ____________________

Roll No: ____________________

Signature: ____________________

Phone Number: ____________________

Qn#	Option	Qn#	Option	Qn#	Option	Qn#	Option	Qn#	Option
1(a)		1(b)		1(c)		1(d)		1(e)	
1(f)		1(g)		1(h)		1(i)		1(j)	

Table 2: Question1:MCQ Answer

Qn#	Answer	Qn#	Answer
2(a)	T(n)=θ( )	2(b)	
2(c)	Insertion Sort: θ( ) Quick Sort:θ( )	2(d)	
2(e)	{a, },{d, },{j, },{h, }	2(f)	
2(g)	max_flow= , cut ={ }	2(h)	
2(i)		2(j)	
2(k)		2(l)	
2m(a)	a: ,b: , c: , x: , y: , z:		
2m(b)	a$\to b$: , $c \to a$: , $b \to c$: , $c \to x$: c$\to y$: , $z \to x$: , $z \to y$:	2(n)	

Table 3: Question 2: Fill in the blanks

For Office Use:

correct answer (x)	wrong answer(y)	Total Score(2x-y/2)

Table 4: Question No1

Total Score:

Question No	Full Score	Scored
1	20	
2	30	
3	3	
4	3	
5	4	
All	60	

Table 5: Total Score