SRN □□□□□□□□

PES

PES UNIVERSITY

UE18CS311

In Semester Assessment (ISA1) B. Tech. 5th SEMESTER – Aug - Dec-2020

UE18CS311 - **Advanced Algorithm**

Time: 2 Hrs. Answer All Questions Max Marks: 60

You can bring one or more hand written notes. No printed or photocopied matter is allowed.

Answer precisely and briefly.

1	a	Show that for any integer constants a and b, where b > 0, (n + a) ^ b is theta(n^b).	2
	b	Given the recurrence T(n) = T(n-1) + n - 1, we assume that the rate of growth is O(n). By substitution method, show whether our assumption is valid. T(0) = 0; T(1) = 0	4
	c	Analyze the quicksort algorithm using recursion tree method. Assume that the pivoting divides the array in the ratio 1:9. T(n) = T(n/10) + T(9n/10) + n	4
2	a	I) In accounting method on stack operations(push, pop and multipop), we associate a cost of 2 on push and 0 on the rest. Why not associate 2 with pop and 0 with push? ii) What is the total cost of executing n of the stack operations push , pop , and multipop , assuming that the stack begins with s objects and finishes with t objects? Hint: consider n operations and change in size of stack	2 + 3
	b	Master Theorem for recurrences of the form T(n) = aT(n/b) + n ^ d compares a and b^d and not a and b. What is the significance of b^d compared to b? (State precisely in a couple of lines)	2
	c	Given that problem A is NP complete, we have to show that problem B is also NP complete. Assume that we can reduce A to B or B to A in polynomial time. Should we reduce A to B or B to A. Why? (State precisely in a couple of lines)	1 + 2
3	a	void search(char pat[], char txt[], int q, int d)	1 + 2 + 2

```
{
  int M = strlen(pat);
  int N = strlen(txt);
  int i, j;

  int p = 0; // hash value for pattern
  int t = 0; // hash value for txt
  int h = 1;
  for (i = 0; i < _____ ; i++) // A
      h = (h * d) % q;
  for (i = 0; i < M; i++)
  {
     p = (d * p + pat[i]) % q;
     t = (d * t + txt[i]) % q;
  }
for (i = 0; i <= ______; i++)  // C
{

   if ( p == t )
   {
      for (j = 0; j < M; j++)
      {
       if (txt[i+j] != pat[j])
           break;
      }
      if (j == M)
        cout<<"Pattern found at index "<< i<<endl;
   }

  if ( i < N-M )
   {
      t = (d*(t - txt[i]*h) + txt[i+M])%q;
                  // B
      if (t < 0)
        t = (t + q);
   }
 }
}
```

Answer the following based on this implementation of Rabin Karp.

I) Fill up the blank at A – observe h is used in the last part of this algorithm.

ii) Do we require the selection at B? Justify your answer in one line

iii) Observe code at C. Fill up the blank. Indicate the semantics of the variable i.

b This algorithm finds transition table of a FSM. 2

```
Algo Compute_transition_function(P[1 .. m], Σ )
for q ← 0 to m do
    for each character a in Σ do
       k ← min(m + 1. q + 2)
       repeat
          k <- k – 1
       until $P_k$ is suffix of  $P_qa$
       delta(q, a) ← k
return delta
```

State the significance of why k is set to min of m + 1 and q + 2.

c 3

```
Algo compute_prefix_function(p)
m←length[p]
π[1]←0
k←0
for q←2 to m do
   while k>0 and  p[k+1] not equal p[q] do
        k←π[k]
   if  p[k+1]=p[q] then
```

```
        k←k+1
    π[q]←k
return π
```

It is stated that the complexity of this algorithm is Theta(m).

Argue either for or against this statement. Be very precise.

4 a In a suffix tree of string T, we are checking the # of occurrences of string S. — 3

How do we make out each of these cases?

0 occurrence

1 occurrence

more than 1 occurrence.

Answer in 3 lines.

b The string T has length n. We construct a suffix tree for T$. — 2 + 3

I) state the number of leaves in terms of n. Give your reason. (in one line)

ii) show that the # of nodes in theta(n)

c Each edge in a suffix tree of T of length n represents a sequence of characters from T. The space requirement in that case will be O(n^2). — 2

How can we make the space requirement linear in n.

5 a This is the optimal value to the classical coin row problem. — 2 + 3

Index	0	1	2	3	4	5	6	7
coin		7	9	10	9	3	5	2
total	0	7	9	17	18	20	23	23

Given this find which coins are selected.

Develop an algorithm to display the coins selected given the solution.

b Fill the blanks in bottom up cut-rod dynamic programming algorithm. — 3

Array p[i] has the price of piece of length i.

```
Algo bottomup_cut_rod(p, n)
// r [0 .. n] : array
r[0] ← 0
for j ← 1 to n do
    q ← _______ // A
    for i = 1 to j do
        q ← ____(q, p[i] + _______) // B  C
    r[j] ← q
return r[n]
```

c If all the matrices in matrix chain multiplication are square, then matrix chain multiplication optimization algorithm is not meaningful. State true or false. Give reason. — 2

(answer in at most two lines)

6	a	Algo approx_vertex_cover(G) // G(V, E)
		// G is an undirected graph
		C ← { }
		E' ← E
		while E' not empty do
		    let (u, v) be an arbitrary edge in E'
		    C ← C U {u, v} // union
		    remove from E' every edge incident on u or v
		return C
		
		if A is the set of edges selected in the loop, show that
		|C| = 2|A|
		if C* is the optimal solution, show that
		|C| <= 2|C*|
			2 + 2

	b	What assumptions are made to solve TSP by approximation method?	2
	c	Each of n customers gives a hat to a hat-check person at a restaurant. The hat-check person gives the hats back to the customers in a random order. What is the expected number of customers who get back their own hat?	4