Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

Synoptic RE-Examination
Jan 2019

Max. Marks: 100 | **Duration: 3 Hrs**
Class: S.E. | **Semester: IV**
Course Code: IT41 / CE41 | **Branch: IT/COMP**
Name of the Course: Design And Analysis of Algorithm

Instructions:
(1) **All Questions are Compulsory**
(2) **Draw neat diagrams**
(3) **Assume suitable data if necessary**

Q No.	Question	Max. Mks
Q.1 a)	**Marks Distribution:** Explained all three notation with diagram ---------- 05mks Explained all three notation without diagram-------- 03mks	*05*
Q.1 b)	**Marks Distribution:** • Algorithm= 3mks • Time Complexity Analysis =2mks **Insertion Sort Algorithm:** Step 1 – If it is the first element, it is already sorted. return 1; Step 2 – Pick next element Step 3 – Compare with all elements in the sorted sub-list Step 4 – Shift all the elements in the sorted sub-list that is greater than the value to be sorted Step 5 – Insert the value Step 6 – Repeat until list is sorted **Analysis:** **Best Case:** The best case is when the list is already sorted. In this case, there is only one comparison per iteration through the outer loop, giving a total of $N-1$ comparisons. Thus $B(N) = O(N)$. **Worst Case:** The worst case is when there are the maximum number of comparisons for each of the $N-1$ iterations through the outer loop. This is i comparisons for the ith iteration: $W(N) = \sum_{i=1}^{N-1} i = (N-1)N/2$ Thus $W(N) = O(N^2)$.	*05*

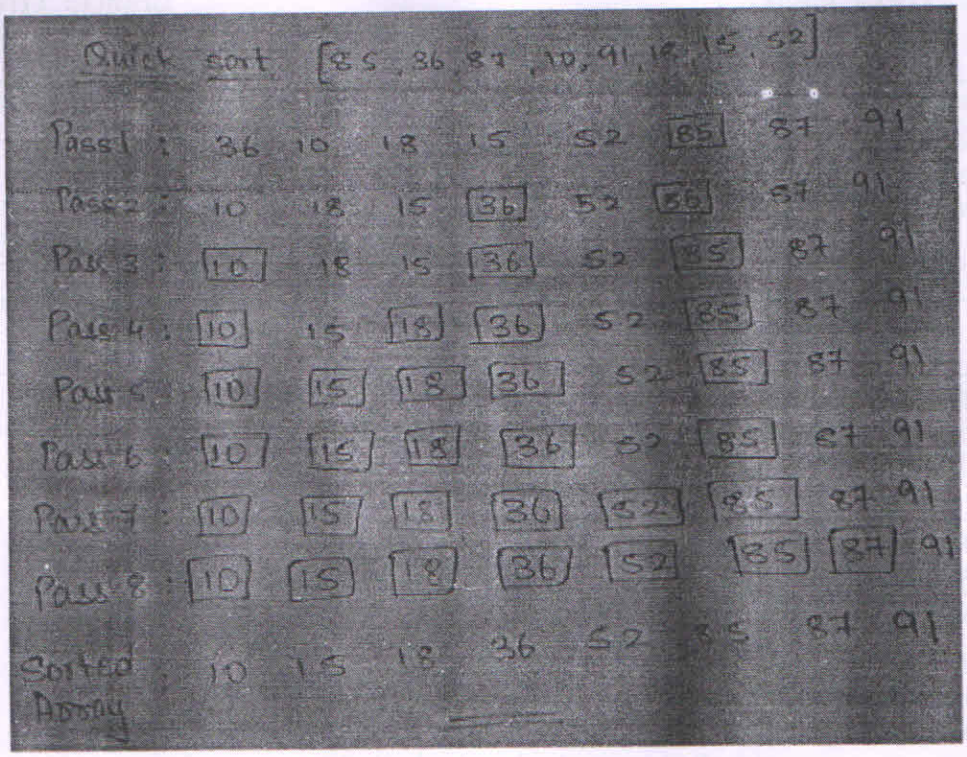
Marks Distribution:

Q.1 c) Solved correctly with all eight passes shown-------- 04mks

Solved correctly with few/ some passes -------(0.5 (half)mk for each correct pass)

Derived the Best and Worst Case time complexity correctly------ 02mks

Only stated the Best and Worst Case Time complexity—0.5(half)mk



OR

Marks Distribution:

Algorithm-----05 marks

Time complexity analysis----------02 marks

Simulation with correct tree drawn----03 marks

```
1   Algorithm MaxMin(i, j, max, min)
2   // a[1 : n] is a global array. Parameters i and j are integers,
3   // 1 ≤ i ≤ j ≤ n. The effect is to set max and min to the
4   // largest and smallest values in a[i : j], respectively.
5   {
6       if (i = j) then max := min := a[i]; // Small(P)
7       else if (i = j − 1) then  // Another case of Small(P)
8           {
9               if (a[i] < a[j]) then
10              {
11                  max := a[j]; min := a[i];
12              }
13              else
14              {
15                  max := a[i]; min := a[j];
16              }
17          }
18          else
19          {   // If P is not small, divide P into subproblems.
20              // Find where to split the set.
21                  mid := ⌊(i + j)/2⌋;
22              // Solve the subproblems.
23                  MaxMin(i, mid, max, min);
24                  MaxMin(mid + 1, j, max1, min1);
25              // Combine the solutions.
26                  if (max < max1) then max := max1;
27                  if (min > min1) then min := min1;
28          }
29  }
```

gorithm 3.6 Recursively finding the maximum and minimum

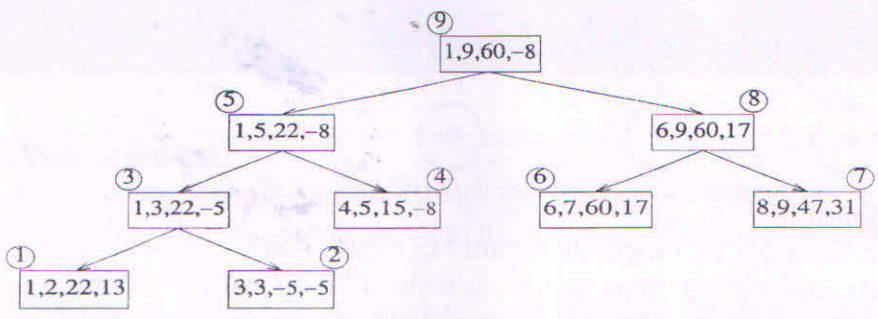
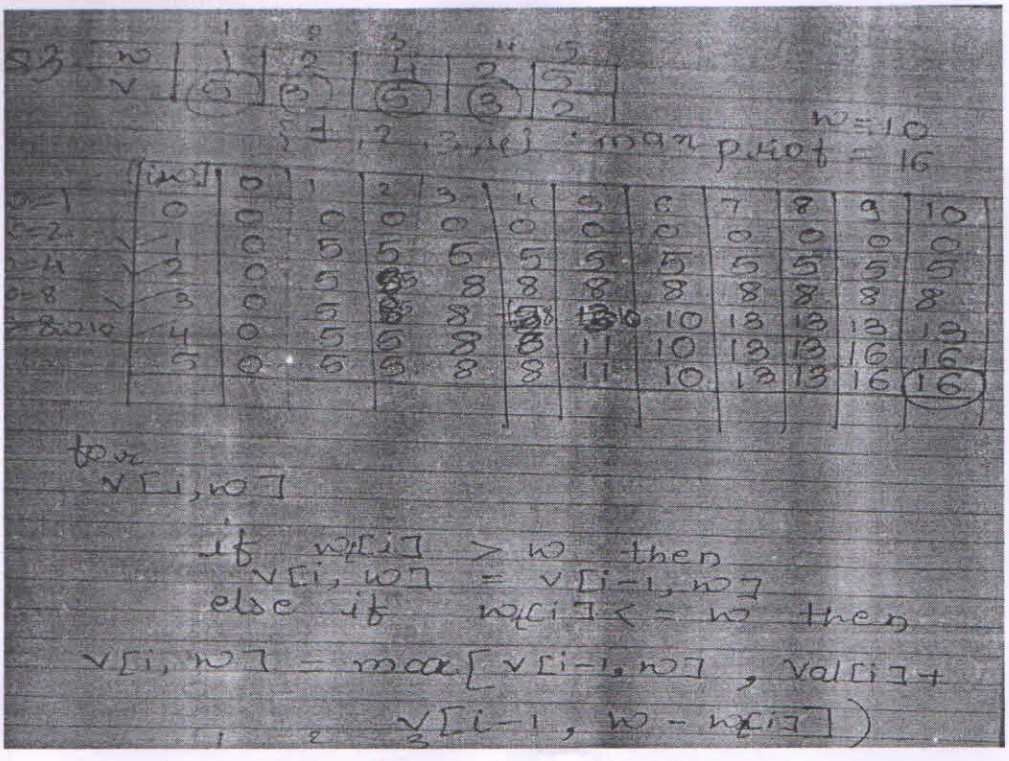
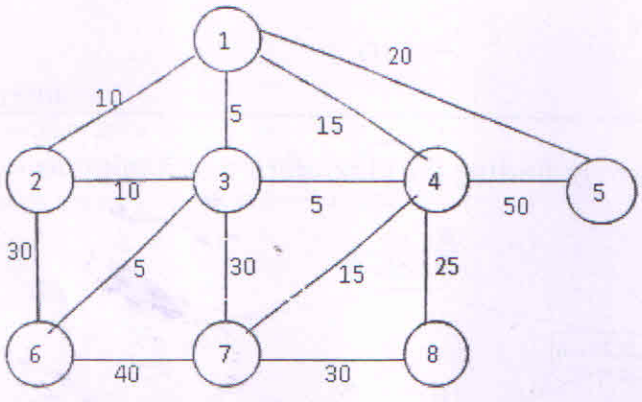
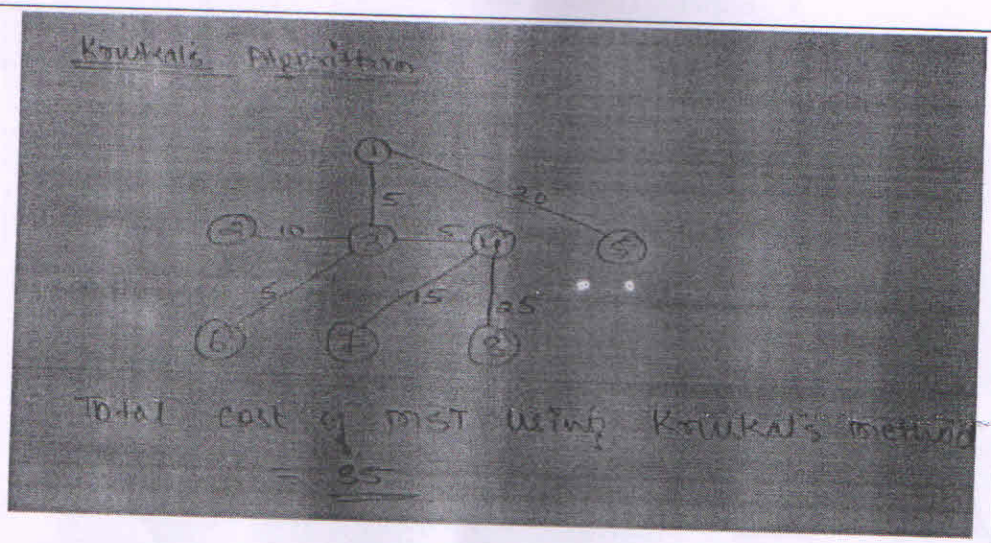
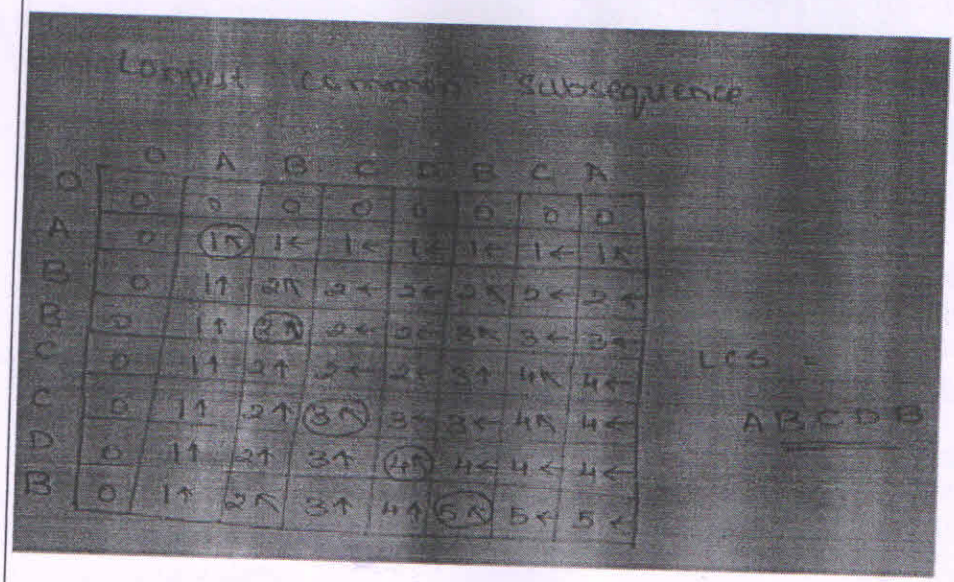
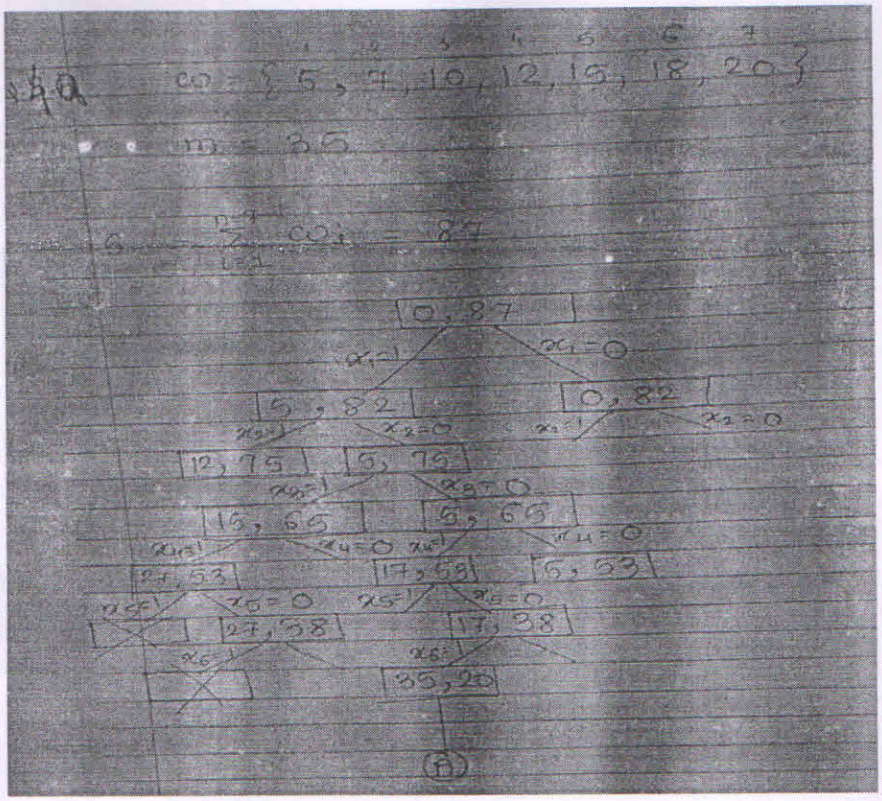
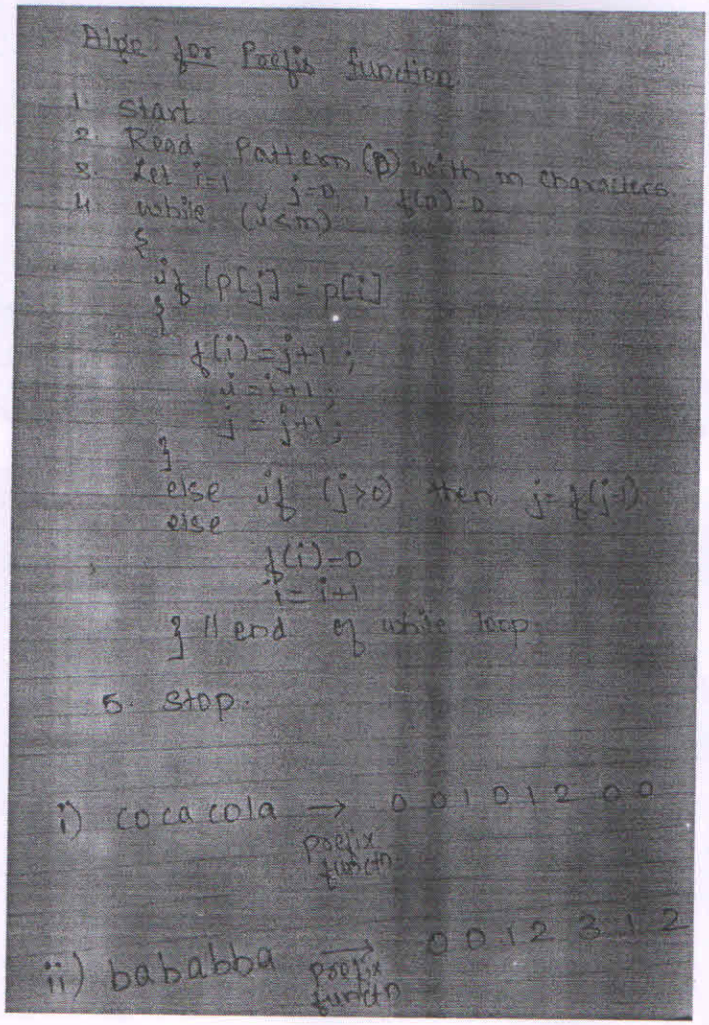


Figure 3.2 Trees of recursive calls of MaxMin

2 a)	 **Marks Distribution:** Partially Solved correctly with formula written correctly ----------05 marks Correctly solved with formula and Matrix ----------10 marks	*10*
Q2. b)	 **Marks Distribution:** • Solved Correctly with steps shown and MST Cost Stated--------4mks • Solved Correctly with steps shown and without MST Cost --------3mks • Solved Correctly without steps shown and MST Cost Stated--------1mk • Solved Correctly without steps shown and without MST Cost --------0.5(half)mk	*10*

		
Q.3 a)	**Marks Distribution:** • Explain LCS using Dynamic Programming Approach.----- 05mks • Solved Correctly------ 5mks 	*10*
Q.3 b)	**Marks Distribution:** • Compare Backtracking and Branch and Bound------ 5mks • Explain Graph coloring problem using Backtracking Approach with the help of State Space Tree---------- 05mks • Explain Graph coloring problem using Backtracking Approach without State Space Tree---------- 03mks **OR** **Marks Distribution:** • Explain Traveling Salesperson problem using Branch and Bound with the help of State Space Tree---------- 10mks	*10*

	• Explain Traveling Salesperson problem using Branch and Bound without State Space Tree---------- 07mks	
Q.4 a)	 **Marks Distribution:** Backtracking algorithm for sum of subset problem----------05 marks Correctly drawn state space tree----------05 marks	*10*
Q4 b)	***Marks Distribution:*** KMP-Prefix Function algorithm -------- **06mks** Correctly solved the prefix function for the given pattern ------- **02mks for each pattern**	*10*



OR

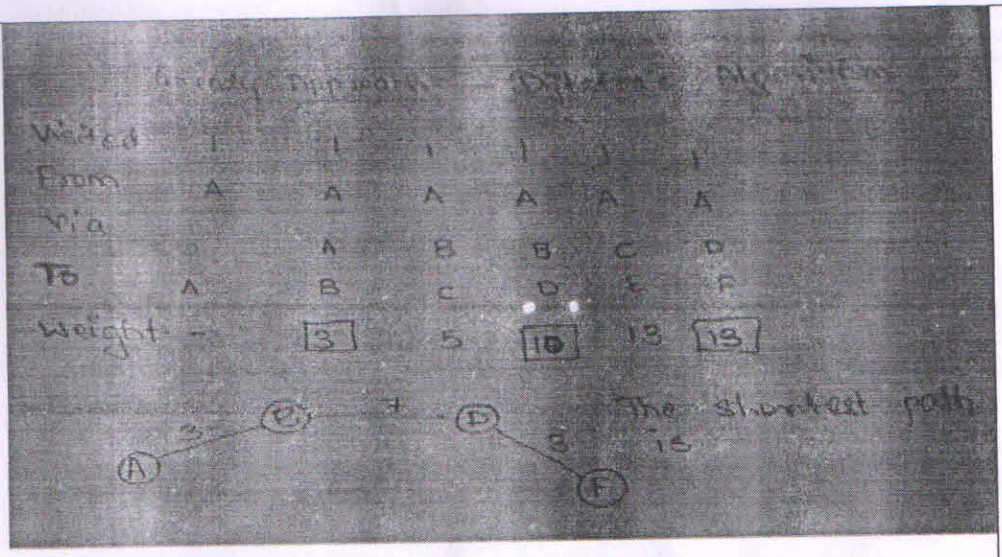
Marks Distribution:

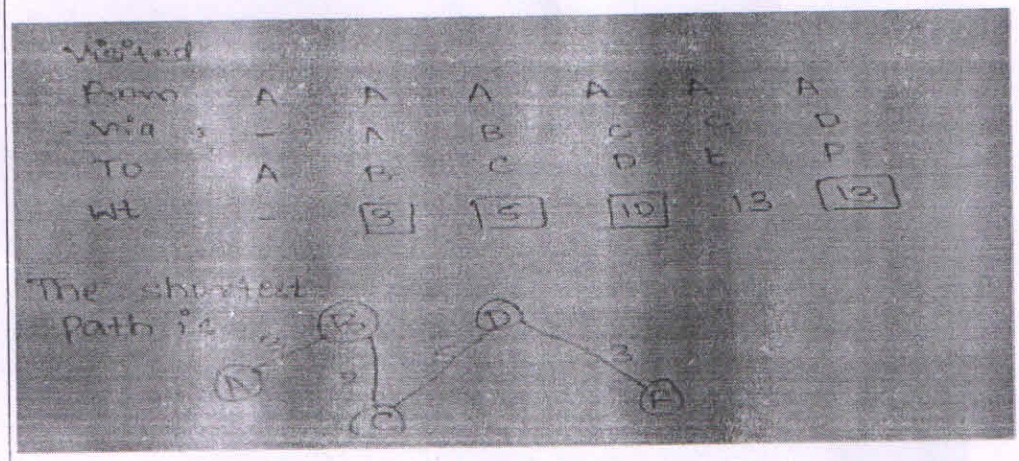
Rabin Karp problem Solved correctly------- 10mks

Q.5a)

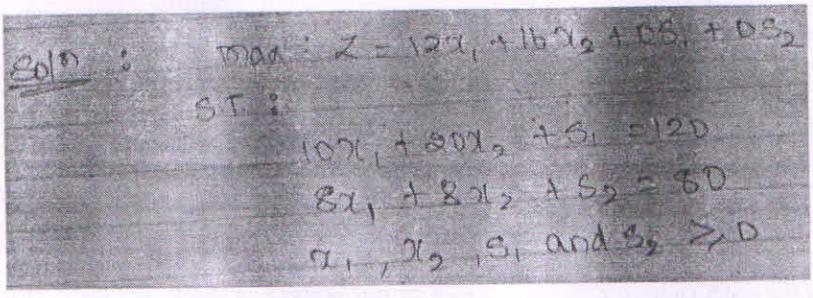
Marks Distribution:

- Solved correctly with priority queue shown at each stage and show the final graph of shortest distance from A to F------ 05mks
- Solved correctly with priority queue shown at each stage and final graph not shown------04mks
- Solved correctly without priority queue shown at each stage and show the final graph of shortest distance from A to F----- 01mk





Q.5.b)	**Marks Distribution:** Solved correctly---------- 10mks	*10*



Initial Simplex Table

	CBi	Cj / Basic Var	12 x_1	16 x_2	0 S_1	0 S_2	Soln	Ratio
	0	S_1	10	20	1	0	120	120/20 = 6
	0	S_2	8	8	0	1	80	8/8 = 10
		Z_j	0	0	0	0	0	
		$C_j - Z_j$	12	16	0	0		

$Z_j = \sum_{i=1}^{2} (CB_i)(a_{ij})$ all $C_j - Z_j \leq 0$ For Max

all $C_j - Z_j \geq 0$ For min

Iteration 1

CBi	Cj / B.V	12 x_1	16 x_2	0 S_1	0 S_2	Soln	Ratio
16	x_2	1/2	1	1/2	0	6	6/(1/2) = 12
0	S_2	(4)	0	-2/5	1	32	32/4 = 8
	Z_j	8	16	4/5	0		
	$C_j - Z_j$	4	0	-4/5	0		

$\Rightarrow$ New value = Old value $-$ (Corr. Key Col. Val. $\times$ Corr. Key Row Value) / Key Element

$8 - \frac{8 \times 10}{20} = 8 - \frac{80}{20} = 8 - 4 = 4$

$8 - \frac{8 \times 20}{20} = 8 - \frac{160}{20} = 8 - 8 = 0$

$1 - \frac{8 \times 0}{20} = 1 - 0 = 1$

$80 - \frac{8 \times 120}{20} = 80 - 960/20 = 80 - 48 = 32$

Iteration II

CBi	Cj	12	16	0	0	Soln	Ratio
	B.V.	x_1	x_2	S_1	S_2		
16	x_2	0	1	1/10	-1/8	2	
12	x_1	1	0	-1/10	1/4	8	
	Zj	12	16	2/5	1	128	
	Cj-Zj	0	0	-2/5	-1		

$1/2 - \left(\frac{1/2 \times 4}{4}\right) = 0$

$1 - \left(\frac{1/2 \times 0}{4}\right) = 1$

$1/20 - \left(\frac{1/2 \times -2/5}{4}\right) = 1/10$

$0 - \left(\frac{1/2 \times 1}{4}\right) = -1/8$

$6 - \left(\frac{1/2 \times 32}{4}\right) = 2$

$\therefore \quad x_1 = 12 \qquad x_2 = 16$

$Z(\text{optimum}) = 128$