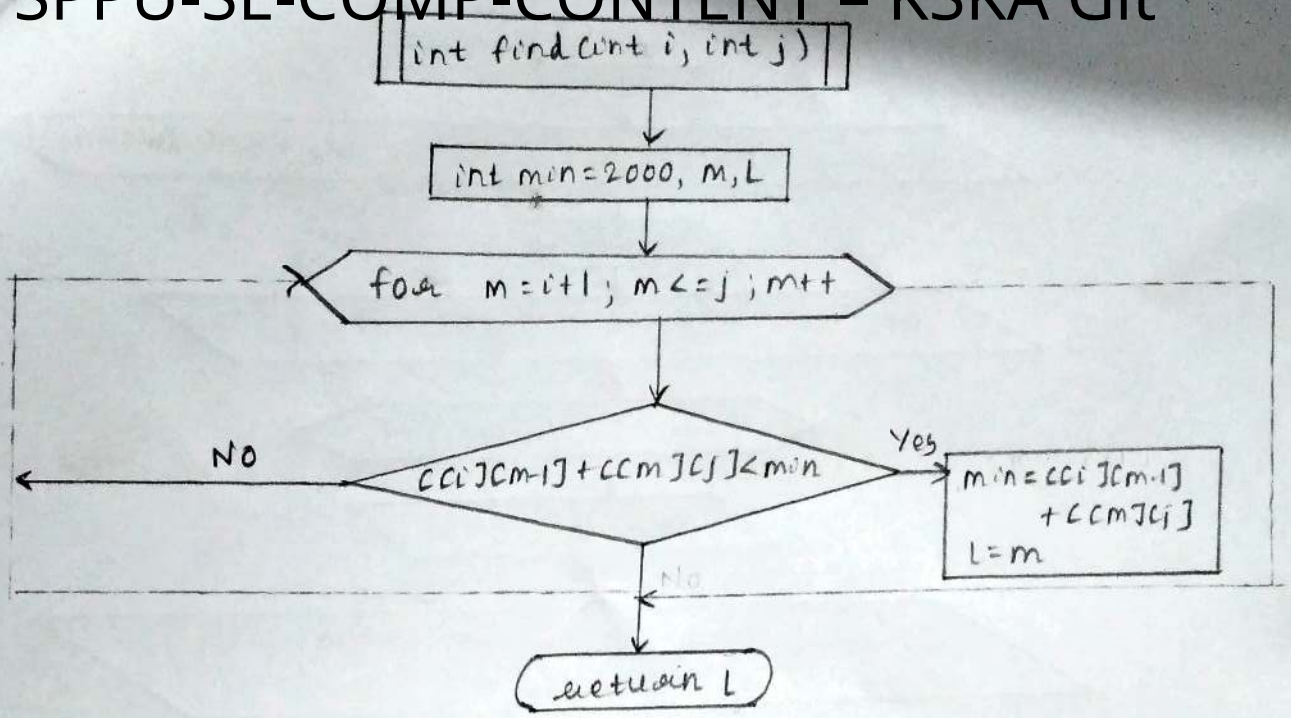
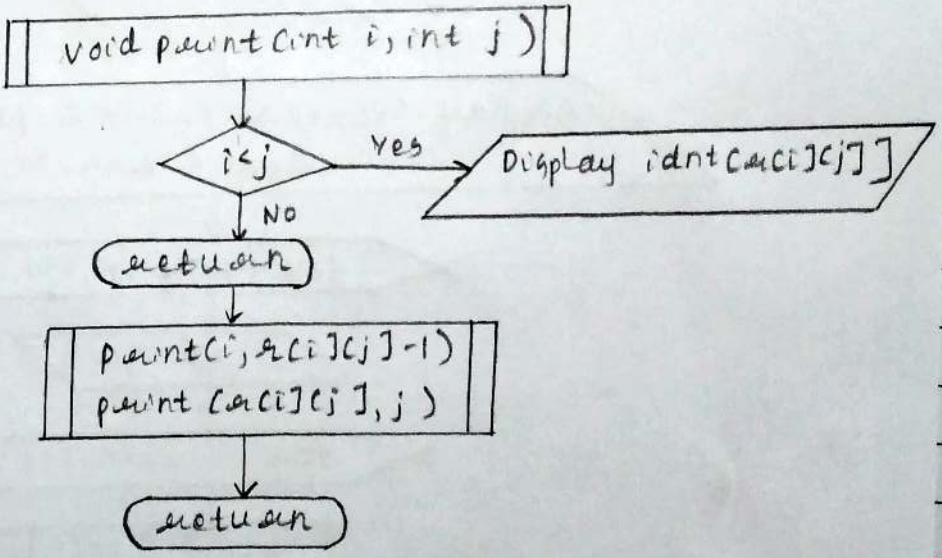
* Optimal Binary Search Tree (OBST)

- An optimal Binary Search Tree (OBST), also known as Weighted Binary Search Tree, is a binary search tree that minimizes the expected search cost.
- In BST, the search cost is the no. of comparisons required to search for a given key.
- In OBST, each node is assigned a weight that represents the probability of the key being searched for.
- The expected search cost of a node is the sum of the product of its depth and weight and the expected search cost of its children.
- Once the OBST is constructed, the time complexity of searching for key is O(log n), the same as for a regular binary search tree.
- The OBST is a useful data structure in applications where the keys have different probabilities of being searched for.
- It can be used to improve the efficiency of searching and retrieval operations in databases, compilers, and other computer programs.

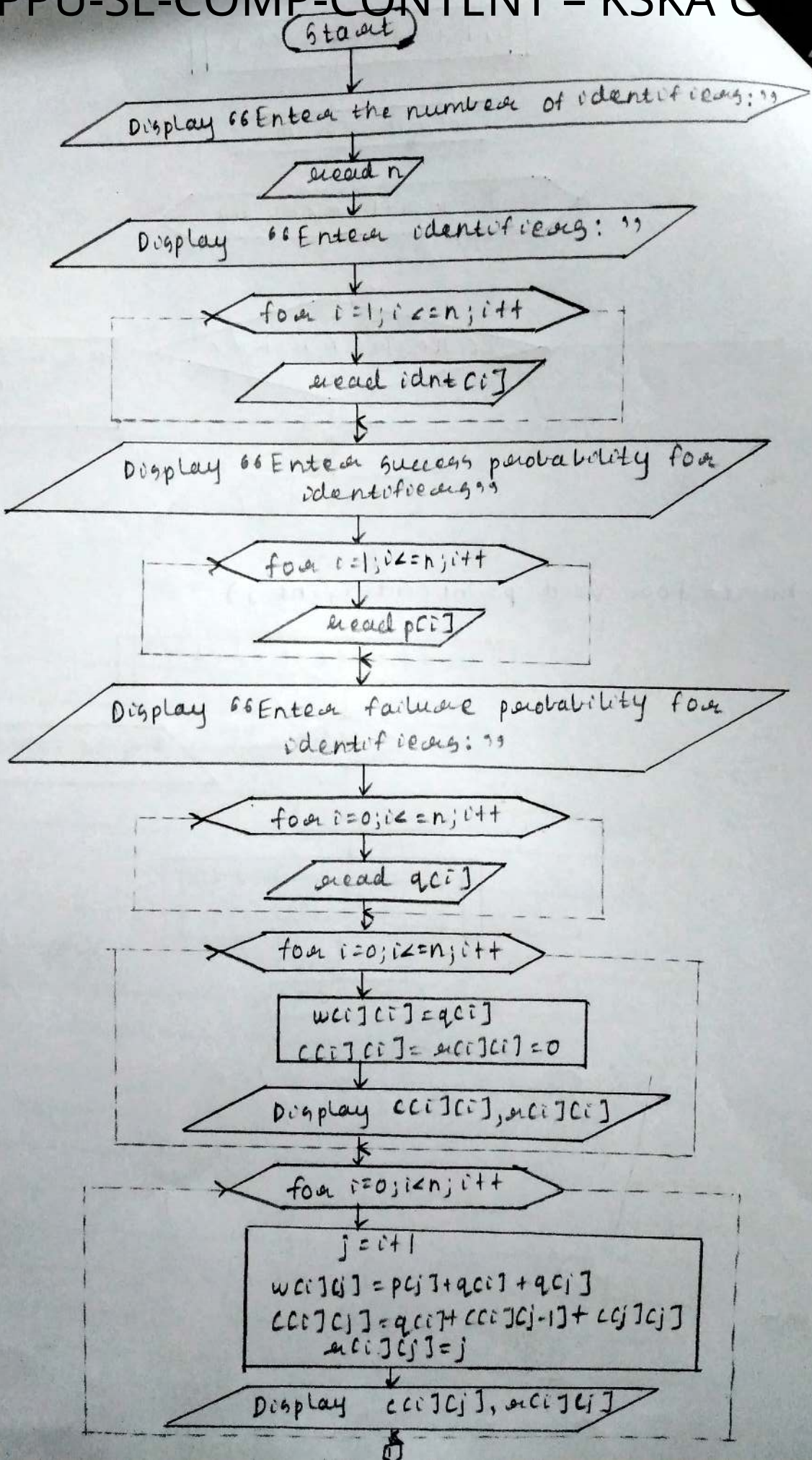
Flowchart for int find(int i, int j)

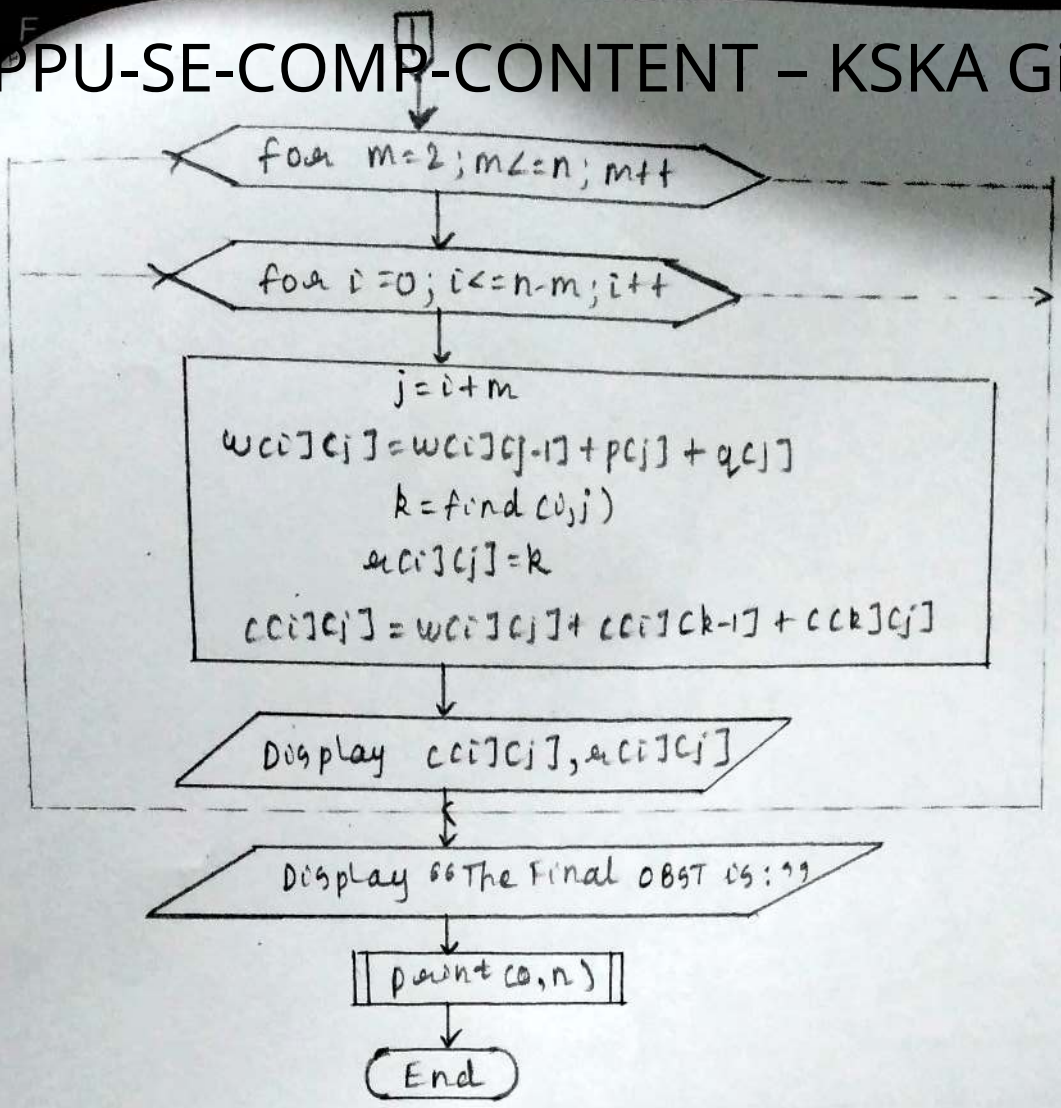


→Flowchart for void print(int i, int j)



→ Flowchart for int main()





→ Pseudocode for int fund (int i, int j)

1. Initialize min = 200
2. Declare int m, L
3. for m = i+1; m <= j; m++ do
   begin
      if C[i][m-1] + a[m][j] < min then
         set min = C[i][j][m-1] + C[C[m][j]
         initialize L = m
   end
4. return L

→ Pseudocode for void print (int i, int j)

1. if i < j then
      Display ident [a[i][j]]
   else
      return
2. call function print (a, a[i][j] - 1)
   call function print (a[a[i][j]], j)
3. return

→ Pseudo code for int main()

1. Start
2. Display "Enter the number of identifiers: "
3. read n
4. for i = 1; i <= n; i++ do
   begin
      read ident[i]
   end
5. Display "Enter success probability for identifiers: "
6. for i = 1; i <= n; i++ do
   begin
      read p[i]
   end

```
7.  Display "Enter failure probability for identifiers"
8.  for i=0 ; i<=n; i++ do
    begin
        read q[i]
    end
9.  for i=0; i<=n; i++ ~~begin~~ do
    ~~do~~ begin
        initialize w[i][i] = q[i]
                   c[i][i] = r[i][i] = 0
        Display w[i][i], c[i][i], r[i][i]
    end
10. for i=0 ; i<n; i++ ~~begin~~ do
    ~~do~~ begin
        initialize j = i+1
        initialize w[i][j] = p[j] + q[i] + q[j]
                   c[i][j] = q[i] + c[i][j-1] + c[j][j]
                   r[i][j] = j
        Display w[i][j], c[i][j], r[i][j]
    end
11. for m=2; m<=n; m++ ~~do~~ ~~to~~ do
    begin
        for i=0; i<=n-m; i++ do
        begin
            initialize j = i+m
                       w[i][j] = w[i][j-1] + p[j] + q[j]
            call function k = find(i,j)
            store r[i][j] = k
            set c[i][j] = w[i][j] + c[i][k-1] + c[k][j]
            Display w[i][j], c[i][j], r[i][j]
        end
    end
12. call function print(0,n)
13. End
```

Q1. Find the optimal binary search tree for $N=6$, having keys $k_1, \ldots k_6$ and weighs $p1=10, p2=3, p3=9, p4=2, p5=0, p6=10$; $q0=5, q1=6, q2=4, q3=4, q4=3, q5=8, q6=0$. The following figure shows the arrays as they would appear after the initialization and their final disposition.

w indicates weighted indicates

C indicates cost

R indicates root

Index	0	1	2	3	4	5	6
k		3	7	10	15	20	25
p		10	3	9	2	0	10
q	5	6	4	4	3	8	0

Ans.

	0	1	2	3	4	5	6
j-i=0	$w_{00}=5$	$w_{11}=6$	$w_{22}=4$	$w_{33}=4$	$w_{44}=3$	$w_{55}=8$	$w_{66}=0$
	$c_{00}=0$	$c_{11}=0$	$c_{22}=0$	$c_{33}=0$	$c_{44}=0$	$c_{55}=0$	$c_{66}=0$
	$r_{00}=0$	$r_{11}=0$	$r_{22}=0$	$r_{33}=0$	$r_{44}=0$	$r_{55}=0$	$r_{66}=0$
j-i=1	$w_{01}=21$	$w_{12}=13$	$w_{23}=17$	$w_{34}=9$	$w_{45}=11$	$w_{56}=18$	
	$c_{01}=21$	$c_{12}=13$	$c_{23}=17$	$c_{34}=9$	$c_{45}=11$	$c_{56}=18$	
	$r_{01}=1$	$r_{12}=2$	$r_{23}=3$	$r_{34}=4$	$r_{45}=5$	$r_{56}=6$	
j-i=2	$w_{02}=28$	$w_{13}=26$	$w_{24}=22$	$w_{35}=17$	$w_{46}=21$		
	$c_{02}=41$	$c_{13}=39$	$c_{24}=31$	$c_{35}=26$	$c_{46}=32$		
	$r_{02}=1$	$r_{13}=3$	$r_{24}=3$	$r_{35}=5$	$r_{46}=6$		
j-i=3	$w_{03}=41$	$w_{14}=31$	$w_{25}=30$	$w_{36}=27$			
	$c_{03}=79$	$c_{14}=53$	$c_{25}=56$	$c_{36}=53$			
	$r_{03}=2$	$r_{14}=3$	$r_{25}=3$	$r_{36}=6$			

$j-i=4$	$w_{04}=46$	$w_{15}=39$	$w_{26}=40$
	$c_{04}=96$	$c_{15}=78$	$c_{26}=89$
	$r_{04}=3$	$r_{15}=3$	$r_{26}=4$
$j-i=5$	$w_{05}=54$	$w_{16}=49$	
	$c_{05}=121$	$c_{16}=115$	
	$r_{05}=3$	$r_{16}=3$	
$j-i=6$	$w_{06}=64$		
	$c_{06}=158$		
	$r_{06}=3$		

2) i) c_{01} $\quad 0<k<=1$

$\therefore k=1$

$c_{01} = \{c(0,0)+c(1,1)\} + w(0,1)$

$= 21$

ii) c_{12} $\quad 1<k<=2$

$\therefore k=2$

$c_{12} = \{c(1,1)+c(2,2)\} + w(1,2)$

$= 13$

3) i) c_{02} $\quad 0<k<=2$

$k=1, k=2$

$c_{02} = \min\{c(0,0)+c(1,2), c(0,1)+c(2,2)\} + w(0,2)$

$= \min\{13, 21\} + 28$

$= 13+28 = 41$

ii) c_{13} $\quad 1<k<=3$

$k=2, k=3$

$c_{13} = \min\{c(1,1)+c(2,3), c(1,2)+c(3,3)\} + w(1,3)$

$= \min\{17, 13\} + 26$

$= 13+26 = 39$

iii) C_{24} $2 < k <= 4$

$k = 3, k = 4$

$C_{24} = \min\{C(2,2) + C(3,4), C(2,3) + C(4,4)\} + w(2,4)$

$= \min\{9, 17\} + 22$

$= 9 + 22 = 31$

iv) C_{35} $3 < k <= 5$

$k = 4, k = 5$

$C_{35} = \min\{C(3,3) + C(4,5), C(3,4) + C(5,5)\} + w(3,5)$

$= \min\{11, 9\} + 17$

$= 9 + 17 = 26$

v) C_{46} $4 < k <= 6$

$k = 5, k = 6$

$C_{46} = \min\{C(4,4) + C(5,6), C(4,5) + C(6,6)\} + w(4,6)$

$= \min\{18, 11\} + 21$

$= 11 + 21 = 32$

4) i) C_{03} $0 < k <= 3$

$k = 1, k = 2, k = 3$

$C_{03} = \min\{C(0,0) + C(1,3), C(0,1) + C(2,3), C(0,2) + C(3,3)\} + w(0,3)$

$= \min\{39, 21 + 17, 41\} + 41$

$= 38 + 41$

$= 79$

ii) C_{14} $1 < k <= 4$

$k = 2; k = 3, k = 4$

$C_{14} = \min\{C(1,1) + C(2,4), C(1,2) + C(3,4), C(1,3) + C(4,4)\} + w(1,4)$

$= \min\{31, 13 + 9, 39\} + 31$

$= 22 + 31$

$= 53$

iii) C_{25} $2 < k <= 5$

$k = 3, k = 4, k = 5$

$C_{26} = \min\{C(2,2) + C(3,5), C(2,3) + C(4,5), C(2,4) + C(5,5)\} + w(2,5)$

$= \min\{26, 17+31, 31+0\} + 30$

$= 26 + 30$

$= 56$

iv) C_{36} $\quad 3 < k <= 6$

$k = 4, k = 5, k = 6$

$C_{36} = \min\{C(3,3) + C(4,6), C(3,4) + C(5,6), C(3,5) + C(6,6)\} + w(3,6)$

$= \min\{32, 9+18, 26\} + 27$

$= 26 + 27$

$= 53$

5) i) C_{04} $\quad 0 < k <= 4$

$k = 1, k = 2, k = 3, k = 4$

$C_{04} = \min\{C(0,0) + C(1,4), C(0,1) + C(2,4), C(0,2) + C(3,4), C(0,3) + C(4,4)\} + w(0,4)$

$= \min\{53, 21+31, 41+9, 79\} + 46$

$= 50 + 46 = 96$

ii) C_{15} $\quad 1 < k <= 5$

$k = 2; k = 3; k = 4; k = 5$

$C_{15} = \min\{C(1,1) + C(2,5), C(1,2) + C(3,5), C(1,3) + C(4,5), C(1,4) + C(5,5)\} + w(1,5)$

$= \min\{56, 13+26, 39+11, 53\} + 39$

$= 39 + 39 = 78$

iii) C_{26} $\quad 2 < k <= 6$

$k = 3; k = 4; k = 5; k = 6$

$C_{26} = \min\{C(2,2) + C(3,6), C(2,3) + C(4,6), C(2,4) + C(5,6), C(2,5) + C(6,6)\} + w(2,6)$

$= \min\{53, 17+32, 31+18, 56\} + 40$

$= 49 + 40 = 89$

6) i) C_{05} $0 < k <= 5$

$k = 1, 2, 3, 4, 5$

$C_{05} = \min\{C(0,0)+C(1,5), C(0,1)+C(2,5), C(0,2)+C(3,5), C(0,3)+C(4,5), C(0,4)+C(5,5)\} + w(0,5)$

$= \min\{78, 21+56, 41+26, 79+11, 96\} + 54$

$= 67 + 54 = 121$

ii) C_{16} $1 < k <= 6$

$k = 2, 3, 4, 5, 6$

$C_{16} = \min\{C(1,1)+C(2,6), C(1,2)+C(3,6), C(1,3)+C(4,6), C(1,4)+C(5,6), C(1,5)+C(6,6)\} + w(1,6)$

$= \min\{89, 13+53, 39+32, 53+18, 78\} + 49$

$= 66 + 49$

$= 115$

7) C_{06} $0 < k <= 6$

$k = 1, 2, 3, 4, 5, 6$

$C_{06} = \min\{C(0,0)+C(1,6), C(0,1)+C(2,6), C(0,2)+C(3,6), C(0,3)+C(4,6), C(0,4)+C(5,6), C(0,5)+C(6,6)\} + w(0,6)$

$= \min\{115, 21+89, 41+53, 79+32, 96+18, 121\} + 64$

$= \cancel{\min\{9\}}$ $94 + 64$

$= 158$