Optimal Binary Search Trees (OBST)

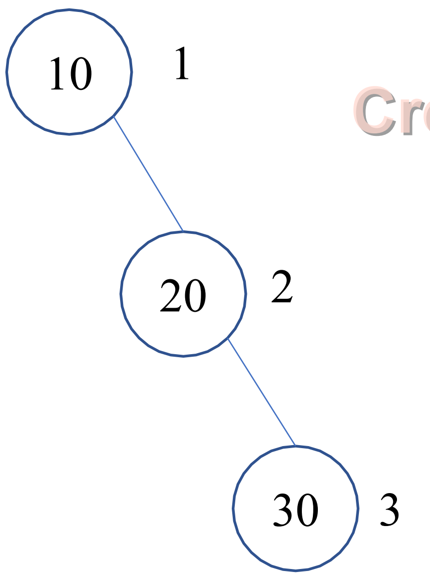
- Suppose we are searching a word from a dictionary.
- For every required word we are looking up in the dictionary then it becomes time consuming process.
- To perform this look up more efficiently we can build the binary search tree of common words as key element.
- We can make this binary search tree efficient by arranging frequently used words nearer to the root and less frequently words away from the root.
- Such a binary search tree makes our task more simplified as well as efficient.
- This type of binary search tree is also called optimal binary search tree (OBST).

Optimal Binary Search Trees (OBST)

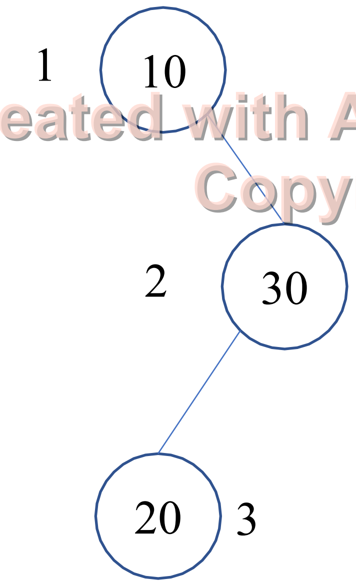
- Let $\{a_1, a_2, a_3, \ldots\ldots, a_n\}$ be a set of keys such that $a_1 < a_2 < a_3$.
- Let p(i) be the probability of successful search and q(i) be the probability of unsuccessful search for an key element i.

Optimal Binary Search Trees (OBST)

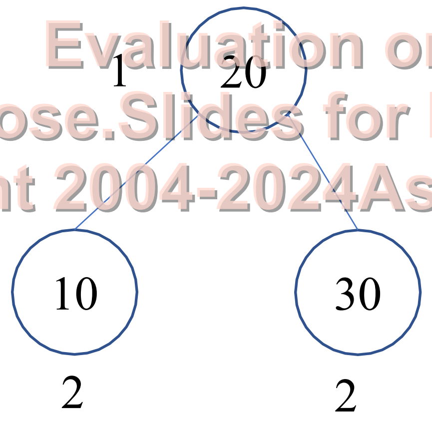
- Let us consider the three key elements 10, 20 and 30.
- The binary search trees for the above keys have been shown below.



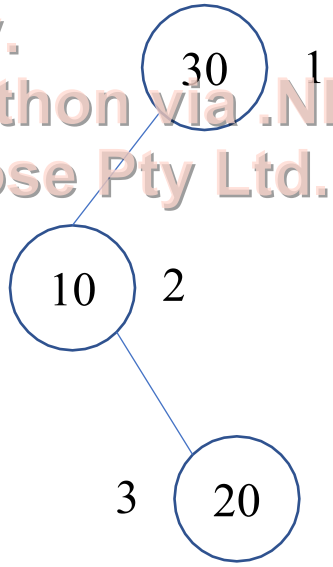
Average No. of Comparisons = (1+2+3)/3 = 6/3 = 2



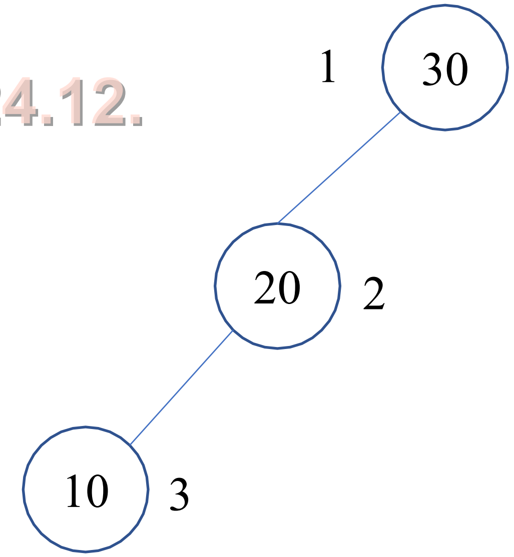
Average No. of Comparisons = (1+2+3)/3 = 6/3 = 2



Average No. of Comparisons = (1+2+2)/3 = 5/3 = 1.6



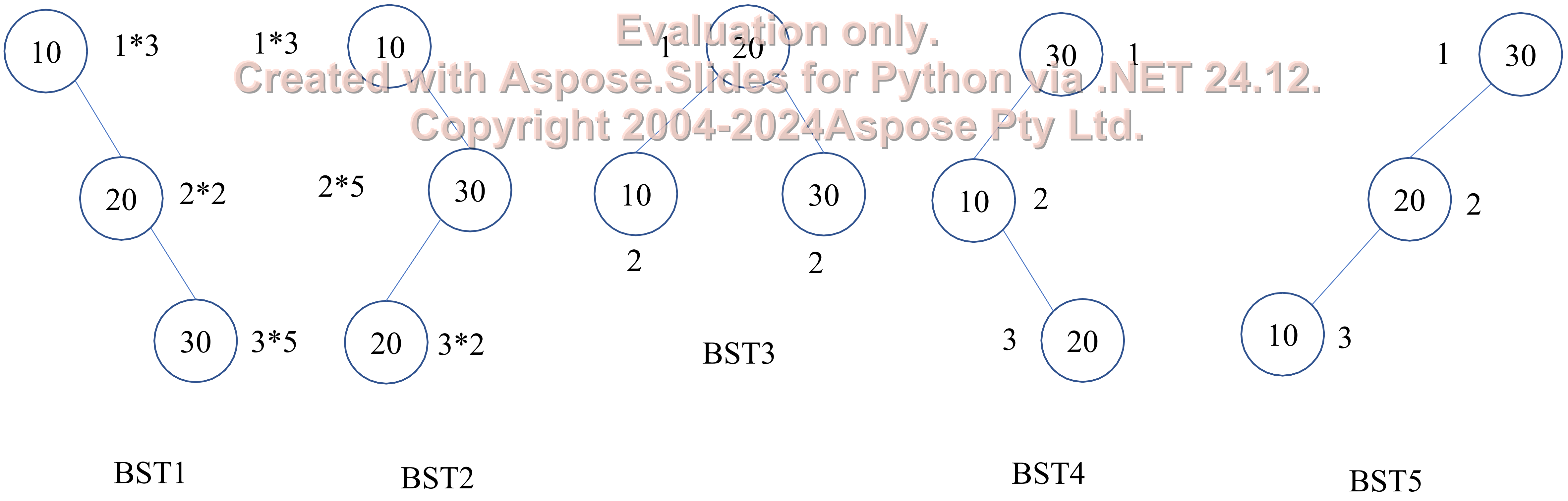
Average No. of Comparisons = (1+2+3)/3 = 6/3 = 2



Average No. of Comparisons = (1+2+3)/3 = 6/3 = 2

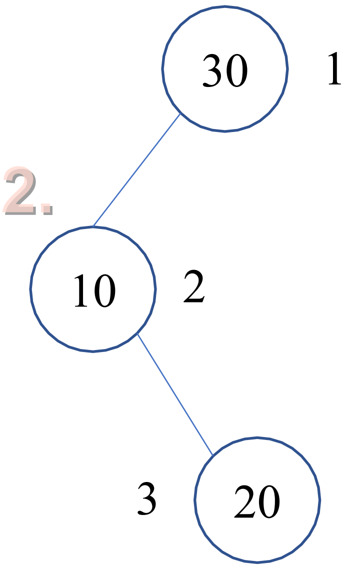
Optimal Binary Search Trees (OBST)

- Let us consider the three key elements 10, 20 and 30.
- Let the frequencies associated with the key elements are 3, 2 and 5 respectively.
- The binary search trees for the above keys have been shown below.



Optimal Binary Search Trees (OBST)

- By considering the frequency the cost of the Binary search trees have been calculated as below.
- Cost of Binary Search Tree 1 = 3 x 1 + 2 x 2 + 5 x 3 = 22
- Cost of Binary Search Tree 2 = 3 x 1 + 5 x 2 + 2 x 3 = 19
- Cost of Binary Search Tree 3 = 2 x 1 + 3 x 2 + 5 x 2 = 18
- Cost of Binary Search Tree 4 = 5 x 1 + 3 x 2 + 2 x 3 = 17
- Cost of Binary Search Tree 5 = 5 x 1 + 2 x 2 + 3 x 3 = 18
- The Binary Search Tree 4 is having minimum cost.
- Therefore Binary Search Tree 4 has been considered as OBST.



Optimal Binary Search Trees (OBST)

- Example
- Construct the optimal binary search tree for the following data: n = 4, (a1, a2, a3, a4) = (do, if, int, while), p(1:4) = (3, 3, 1, 1) and q(0:4) = 2, 3, 1, 1, 1)
- Solution

- Computation of W, C and r has been carried out using the following formulae.
- $w_{i,\, i} = q_i$
- $r_{i.\, i} = 0$
- $c_{i,\, i} = 0$

Optimal Binary Search Trees (OBST)

- $w_{i,\, i+1} = q_i + q_{i+1} + p_{i+1}$
- $r_{i.\, i+1} = i+1$
- $c_{i,\, i+1} = q_i + q_{i+1} + p_{i+1}$
- $w_{i,\, j} = w_{i,\, j-1} + p_j + q_j$
- $r_{i.\, j} = k$
- $c_{i,j} = \min\limits_{i<k\leq j}\{c_{i,k-1} + c_{k,j}\} + w_{i,j}$
- Construction of the W Table
- The rows of the W table represents the length l of the tree.
- The length l will be given by
- $l = j - i$

W Table

l = j - i ↓ \ i →	0	1	2	3	4
0					
1					
2					
3					
4					

Optimal Binary Search Trees (OBST)

- For length of the tree 0
- $l = j - i = 0$
- For $i = 0$, $j = 0$, $w_{0,0} = q_0 = 2$
- For $i = 1$, $j = 1$, $w_{1,1} = q_1 = 3$
- For $i = 2$, $j = 2$, $w_{2,2} = q_2 = 1$
- For $i = 3$, $j = 3$, $w_{3,3} = q_3 = 1$
- For $i = 4$, $j = 4$, $w_{4,4} = q_4 = 1$

W Table

$i \rightarrow$

$l = j - i$ ↓

	0	1	2	3	4
0	$w_{0,0} =$ 2	$w_{1,1} =$ 3	$w_{2,2} =$ 1	$w_{3,3} =$ 1	$w_{4,4} =$ 1
1					
2					
3					
4					

$$w_{i,\,i} = q_i$$

$$w_{i,\,i+1} = q_i + q_{i+1} + p_{i+1}$$

$$w_{i,\,j} = w_{i,\,j-1} + p_j + q_j$$

Optimal Binary Search Trees (OBST)

- For length of the tree 1
- $l = j - i = 1$
- For $i = 0$, $j = 1$, $w_{0,1} = q_0 + q_1 + p_1 = 8$
- For $i = 1$, $j = 2$, $w_{1,2} = q_1 + q_2 + p_2 = 7$
- For $i = 2$, $j = 3$, $w_{2,3} = q_2 + q_3 + p_3 = 3$
- For $i = 3$, $j = 4$, $w_{3,4} = q_3 + q_4 + p_4 = 3$

W Table

$i \rightarrow$

$l = j - i$ $\downarrow$

	0	1	2	3	4
0	$w_{0,0} = 2$	$w_{1,1} = 3$	$w_{2,2} = 1$	$w_{3,3} = 1$	$w_{4,4} = 1$
1	$w_{0,1} = 8$	$w_{1,2} = 7$	$w_{2,3} = 3$	$w_{3,4} = 3$	
2					
3					
4					

$w_{i,i} = q_i$

$w_{i,i+1} = q_i + q_{i+1} + p_{i+1}$

$w_{i,j} = w_{i,j-1} + p_j + q_j$

Optimal Binary Search Trees (OBST)

- For length of the tree 2
- $l = j - i = 2$
- For $i = 0$, $j=2$, $w_{0,2} = w_{0,1} + p_2 + q_2 = 12$
- For $i = 1$, $j=3$, $w_{1,3} = w_{1,2} + p_3 + q_3 = 9$
- For $i = 2$, $j=4$, $w_{2,4} = w_{2,3} + p_4 + q_4 = 5$
- For length of the tree 3
- $l = j - i = 3$
- For $i = 0$, $j=3$, $w_{0,3} = w_{0,2} + p_3 + q_3 = 14$
- For $i = 1$, $j=4$, $w_{1,4} = w_{1,3} + p_4 + q_4 = 11$

W Table

$i \rightarrow$

$l = j - i$ $\downarrow$

	0	1	2	3	4
0	$w_{0,0} = 2$	$w_{1,1} = 3$	$w_{2,2} = 1$	$w_{3,3} = 1$	$w_{4,4} = 1$
1	$w_{0,1} = 8$	$w_{1,2} = 7$	$w_{2,3} = 3$	$w_{3,4} = 3$	
2	$w_{0,2} = 12$	$w_{1,3} = 9$	$w_{2,4} = 5$		
3	$w_{0,3} = 14$	$w_{1,4} = 11$			
4					

$w_{i,j} = w_{i,j-1} + p_j + q_j$

Optimal Binary Search Trees (OBST)

- For length of the tree 4
- $l = j - i = 4$
- For $i = 0$, $j=4$, $w_{0,4} = w_{0,3} + p_4 + q_4 = 16$
- Computation of C and r Table

W Table

$i \rightarrow$

$l = j - i$ ↓

	0	1	2	3	4
0	$w_{0,0} =$ 2	$w_{1,1} =$ 3	$w_{2,2} =$ 1	$w_{3,3} =$ 1	$w_{4,4} =$ 1
1	$w_{0,1} =$ 8	$w_{1,2} =$ 7	$w_{2,3} =$ 3	$w_{3,4} =$ 3	
2	$w_{0,2} =$ 12	$w_{1,3} =$ 9	$w_{2,4} =$ 5		
3	$W_{0,3} =$ 14	$w_{1,4} =$ 11			
4	$w_{0,4} =$ 16				

$w_{i,j} = w_{i,j-1} + p_j + q_j$

Optimal Binary Search Trees (OBST)

- Computation of C and r Table
- For length of the tree 0
- $l = j - i = 0$
- For $i = 0$, $j=0$, $c_{0,0} = 0$, $r_{0,0} = 0$
- For $i = 1$, $j=1$, $c_{1,1} = 0$, $r_{1,1} = 0$
- For $i = 2$, $j=2$, $c_{2,2} = 0$, $r_{2,2} = 0$
- For $i = 3$, $j=3$, $c_{3,3} = 0$, $r_{3,3} = 0$
- For $i = 4$, $j=4$, $c_{4,4} = 0$, $r_{4,4} = 0$

C and r Table

$i \rightarrow$

$l = j - i$ $\downarrow$

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1					
2					
3					
4					

$r_{i.\ i} = 0$

$c_{i,\ i} = 0$

Optimal Binary Search Trees (OBST)

- For length of the tree 1
- $l = j - i = 1$
- For $i = 0$, $j=1$, $c_{0,1} = q_0 + q_1 + p_1 = 8$
- For $i = 1$, $j=2$, $c_{1,2} = q_1 + q_2 + p_2 = 7$
- For $i = 2$, $j=3$, $c_{2,3} = q_2 + q_3 + p_3 = 3$
- For $i = 3$, $j=4$, $c_{3,4} = q_3 + q_4 + p_4 = 3$

C and r Table

$i \rightarrow$

$l = j - i$ ↓

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$	$c_{1,2} = 7$	$c_{2,3} = 3$	$c_{3,4} = 3$	
2					
3					
4					

$r_{i.\ i+1} = i+1$

$c_{i,\ i+1} = q_i + q_{i+1} + p_{i+1}$

Optimal Binary Search Trees (OBST)

- For length of the tree 1
- $l = j - i = 1$
- For $i = 0$, $j=1$, $r_{0,1} = 0 + 1 = 1$
- For $i = 1$, $j=2$, $r_{1,2} = 1 + 1 = 2$
- For $i = 2$, $j=3$, $r_{2,3} = 2 + 1 = 3$
- For $i = 3$, $j=4$, $r_{3,4} = 3 + 1 = 4$

C and r Table

$i \rightarrow$

$l = j - i$ $\downarrow$

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2					
3					
4					

$$r_{i.\ i+1} = i+1$$

$$c_{i,\ i+1} = q_i + q_{i+1} + p_{i+1}$$

Optimal Binary Search Trees (OBST)

- For length of the tree 2
- $l = j - i = 2$
- For $i = 0$, $j=2$
- $c_{0,2} = w_{0,2} + \min\limits_{0<k\leq 2} \begin{cases} k = 1, & (c_{0,0} + c_{1,2}) \\ k = 2, & (c_{0,1} + c_{2,2}) \end{cases}$
- $c_{0,2} = 12 + \min\limits_{0<k\leq 2} \begin{cases} k = 1, & (0 + 7) \\ k = 2, & (8 + 0) \end{cases}$
- $c_{0,2} = 12 + 7 = 19$
- $r_{0,2} = k = 1$

C and r Table

$l = j - i = 1$ ↓

i →	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$				
3					
4					

$$c_{i,j} = \min_{i<k\leq j} \{ c_{i,k-1} + c_{k,j} \} + w_{i,j}$$

$r_{i,j} = k$

Optimal Binary Search Trees (OBST)

- For length of the tree 2
- l = j – i = 2
- For i = 1, j=3
- $c_{1,3} = w_{1,3} + \min_{1<k\leq 3} \begin{cases} k=2, & (c_{1,1} + c_{2,3}) \\ k=3, & (c_{1,2} + c_{3,3}) \end{cases}$
- $c_{1,3} = 9 + \min_{0<k\leq 2} \begin{cases} k=2, & (0 + 3) \\ k=3, & (7 + 0) \end{cases}$
- $c_{1,3} = 9 + 3 = 12$
- $r_{1,3} = k = 2$

C and r Table

i →

l = j - i

↓

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$	$c_{1,3} = 12$ $r_{1,3} = 2$			
3					
4					

Optimal Binary Search Trees (OBST)

- For length of the tree 2
- $l = j - i = 2$
- For $i = 2$, $j=4$
- $c_{2,4} = w_{2,4} + \min\limits_{2<k\le 4} \begin{cases} k = 3, & (c_{2,2} + c_{3,4}) \\ k = 4, & (c_{2,3} + c_{4,4}) \end{cases}$
- $c_{2,4} = 5 + \min\limits_{0<k\le 2} \begin{cases} k = 3, & (0 + 3) \\ k = 4, & (3 + 0) \end{cases}$
- $c_{2,4} = 5 + 3 = 8$
- $r_{2,4} = k = 3$

C and r Table

$i \rightarrow$

$l = j - i$ ↓

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2}= 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3					
4					

Optimal Binary Search Trees (OBST)

- For length of the tree 3
- $l = j - i = 3$
- For $i = 0$, $j=3$

$$c_{0,3} = w_{0,3} + \min_{0<k\leq 3} \begin{cases} k = 1, & (c_{0,0} + c_{1,3}) \\ k = 2, & (c_{0,1} + c_{2,3}) \\ k = 3, & (c_{0,2} + c_{3,3}) \end{cases}$$

- $$c_{0,3} = 14 + \min_{0<k\leq 3} \begin{cases} k = 1, & (0 + 12) \\ k = 2, & (8 + 3) \\ k = 3, & (19 + 0) \end{cases}$$

- $c_{0,3} = 14 + 11 = 25$
- $r_{0,3} = k = 2$

C and r Table

$i \rightarrow$

$l = j - i$ $\downarrow$

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3} = 2$				
4					

Optimal Binary Search Trees (OBST)

- For length of the tree 3
- $l = j - i = 3$
- For $i = 1$, $j=4$

$$c_{1,4} = w_{1,4} + \min_{1<k\leq 4} \begin{cases} k = 2, & (c_{1,1} + c_{2,4}) \\ k = 3, & (c_{1,2} + c_{3,4}) \\ k = 4, & (c_{1,3} + c_{4,4}) \end{cases}$$

- $$c_{1,4} = 11 + \min_{1<k\leq 4} \begin{cases} k = 2, & (0 + 8) \\ k = 3, & (7 + 3) \\ k = 4, & (12 + 0) \end{cases}$$
- $c_{1,4} = 11 + 8 = 19$
- $r_{1,4} = k = 2$

C and r Table

i →

l = j - i ↓

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2}= 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3}= 2$	$c_{1,4} = 19$ $r_{1,4} = 2$			
4					

Optimal Binary Search Trees (OBST)

- For length of the tree 4
- l = j – i = 4
- For i = 0, j=4

$$c_{0,4} = w_{0,4} + \min_{0<k\leq 4} \begin{cases} k=1, & (c_{0,0} + c_{1,4}) \\ k=2, & (c_{0,1} + c_{2,4}) \\ k=3, & (c_{0,2} + c_{3,4}) \\ k=4, & (c_{0,3} + c_{4,4}) \end{cases}$$

$$c_{0,4} = 16 + \min_{0<k\leq 4} \begin{cases} k=1, & (0 + 19) \\ k=2, & (8 + 8) \\ k=3, & (19 + 3) \\ k=4, & (25 + 0) \end{cases}$$

- $c_{0,4} = 16 + 16 = 32$
- $r_{0,4} = k = 2$

C and r Table

l = j - i ↓ / i →	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3} = 2$	$C_{1,4} = 19$ $r_{1,4} = 2$			
4	$c_{0,4} = 32$ $r_{0,4} = 2$				

Optimal Binary Search Trees (OBST)

- The root of the Binary Search Tree will be given by $r_{0,n}$.
- From the C and r table $r_{0,4} = 2$.
- Therefore a_2 becomes the root of the BST.
- $r_{i,\,k-1}$ becomes the left child.
- $r_{k,j}$ becomes the right child.

- The binary search tree can be constructed using the above formulation has been shown below.

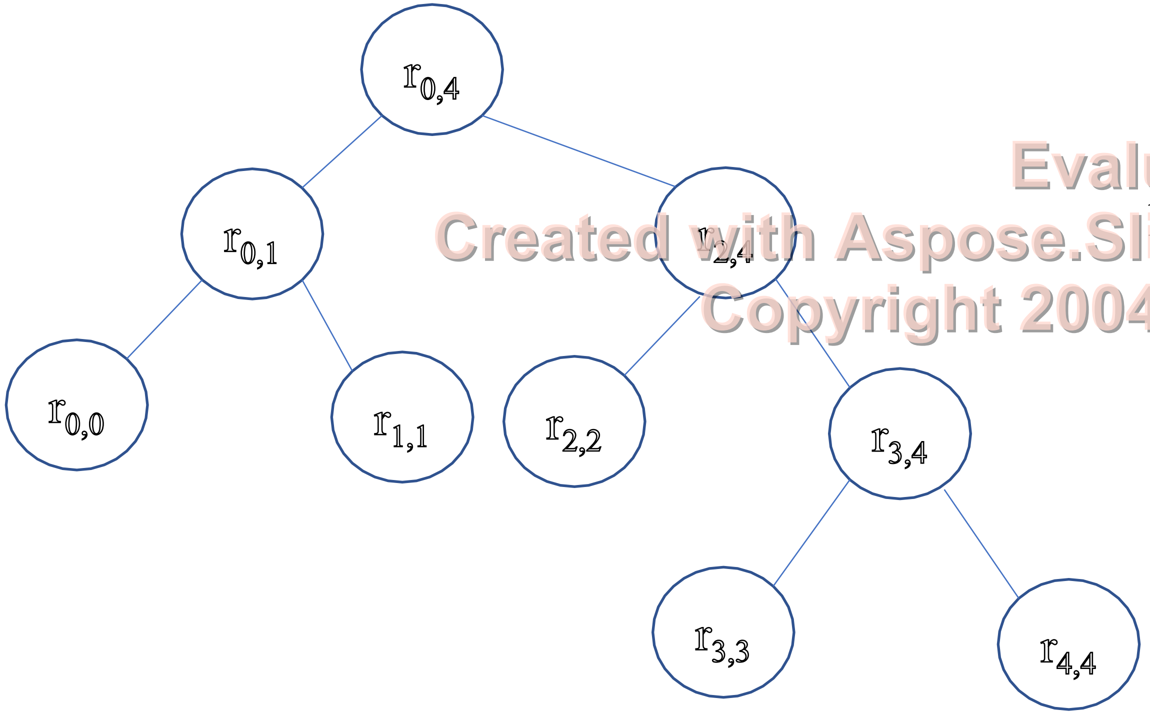
C and r Table

i →

l = j - i ↓

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3} = 2$	$C_{1,4} = 19$ $r_{1,4} = 2$			
4	$c_{0,4} = 32$ $r_{0,4} = 2$				

Optimal Binary Search Trees (OBST)

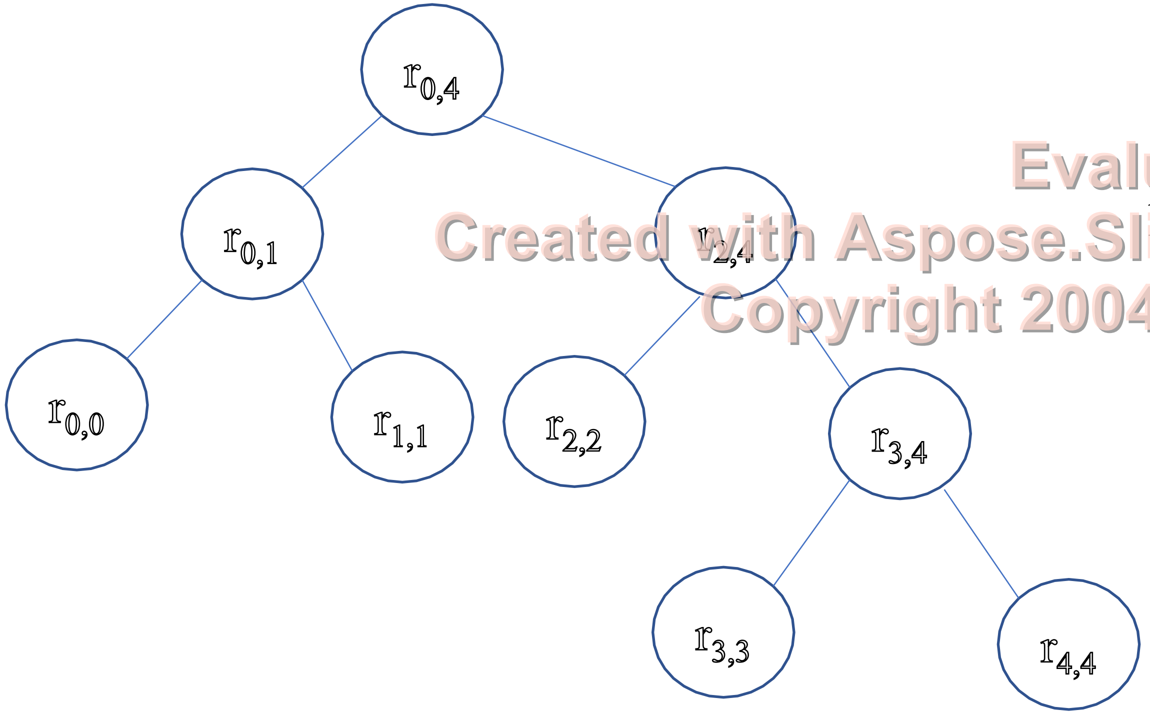


C and r Table

$i \rightarrow$

$l = j - i \downarrow$	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2}= 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3}= 2$	$C_{1,4} = 19$ $r_{1,4} = 2$			
4	$c_{0,4} = 32$ $r_{0,4}= 2$				

Optimal Binary Search Trees (OBST)



C and r Table

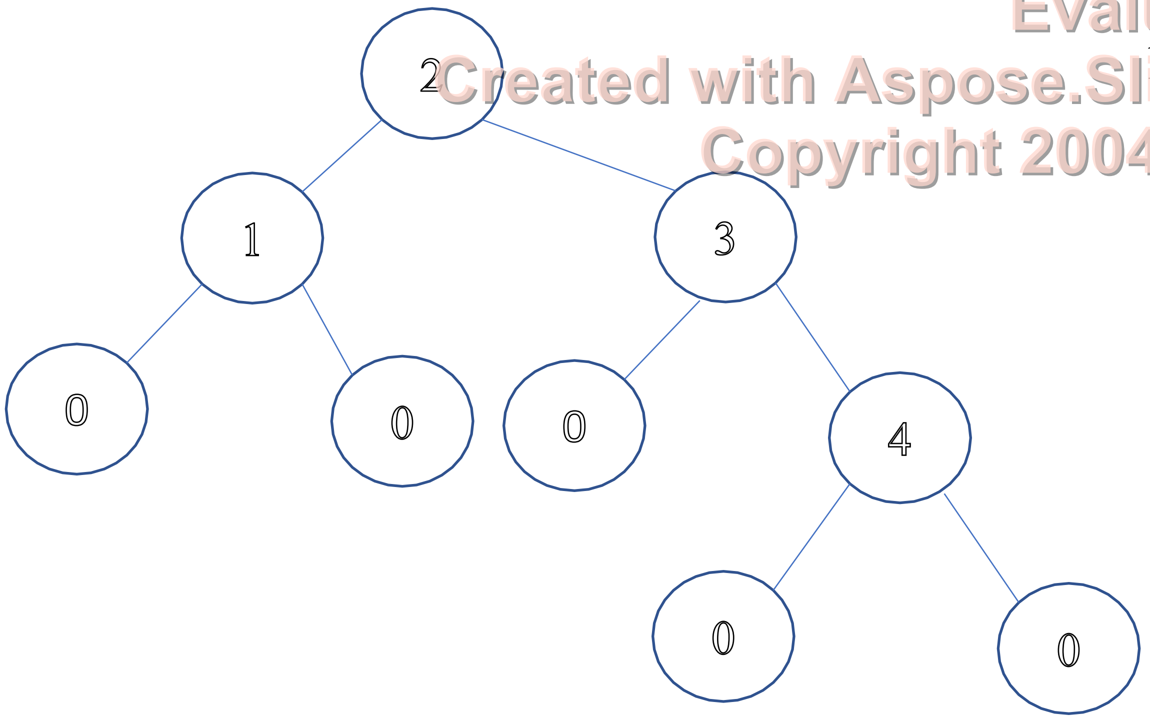
i →

l = j - i ↓

l \ i	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3} = 2$	$C_{1,4} = 19$ $r_{1,4} = 2$			
4	$c_{0,4} = 32$ $r_{0,4} = 2$				

Optimal Binary Search Trees (OBST)

Substituting the root table element values we obtain the following Binary search Tree.



C and r Table

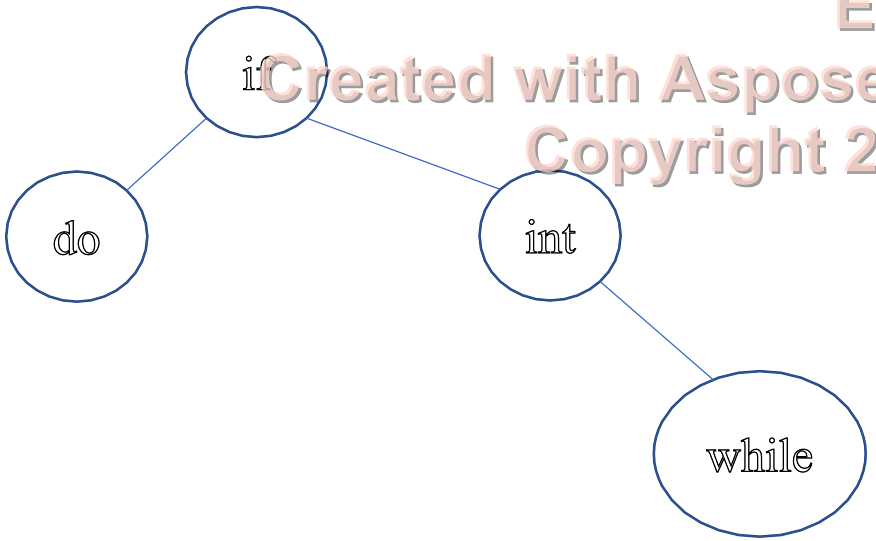
$i \rightarrow$

$l = j - i$ ↓

	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3} = 2$	$C_{1,4} = 19$ $r_{1,4} = 2$			
4	$c_{0,4} = 32$ $r_{0,4} = 2$				

Optimal Binary Search Trees (OBST)

Substituting the corresponding the key elements we obtain the following Optimal Binary search Tree.



(a1, a2, a3, a4) = (do, if, int, while)

C and r Table

i →

l = j - i ↓

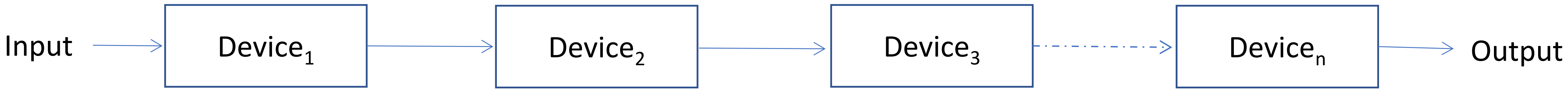
	0	1	2	3	4
0	$c_{0,0} = 0$ $r_{0,0} = 0$	$c_{1,1} = 0$ $r_{1,1} = 0$	$c_{2,2} = 0$ $r_{2,2} = 0$	$c_{3,3} = 0$ $r_{3,3} = 0$	$c_{4,4} = 0$ $r_{4,4} = 0$
1	$c_{0,1} = 8$ $r_{0,1} = 1$	$c_{1,2} = 7$ $r_{1,2} = 2$	$c_{2,3} = 3$ $r_{2,3} = 3$	$c_{3,4} = 3$ $r_{3,4} = 4$	
2	$c_{0,2} = 19$ $r_{0,2} = 1$	$c_{1,3} = 12$ $r_{1,3} = 2$	$c_{2,4} = 8$ $r_{2,4} = 3$		
3	$c_{0,3} = 25$ $r_{0,3} = 2$	$C_{1,4} = 19$ $r_{1,4} = 2$			
4	$c_{0,4} = 32$ $r_{0,4} = 2$				

Optimal Binary Search Trees (OBST)

- Example
- Construct the optimal binary search tree for the following data: n = 4, (a1, a2, a3, a4) = (end, goto, print, stop), with p(1) = 1/20, p(2) = 1/5, p(3) = 1/10, p(4) = 1/20. q(0) = 1/5, q(1) = 1/10, q(2) = 1/5, q(3) = 1/20, q(4) = 1/20.
- Solution
- Computation of W, C and r has been carried out using the following formulae.
- $w_{i,\,i} = q_i$
- $r_{i.\,i} = 0$
- $c_{i,\,i} = 0$

Reliability Design

- The problem is to design a system that is composed of several devices connected in series.
- When devices are connected then it is necessary that each device should work properly.
- The probability that a device i will work properly is called as reliability of the device.
- Let r_i be the reliability of the device D_i.
- The reliability of the entire system will be given by $\prod r_i$.

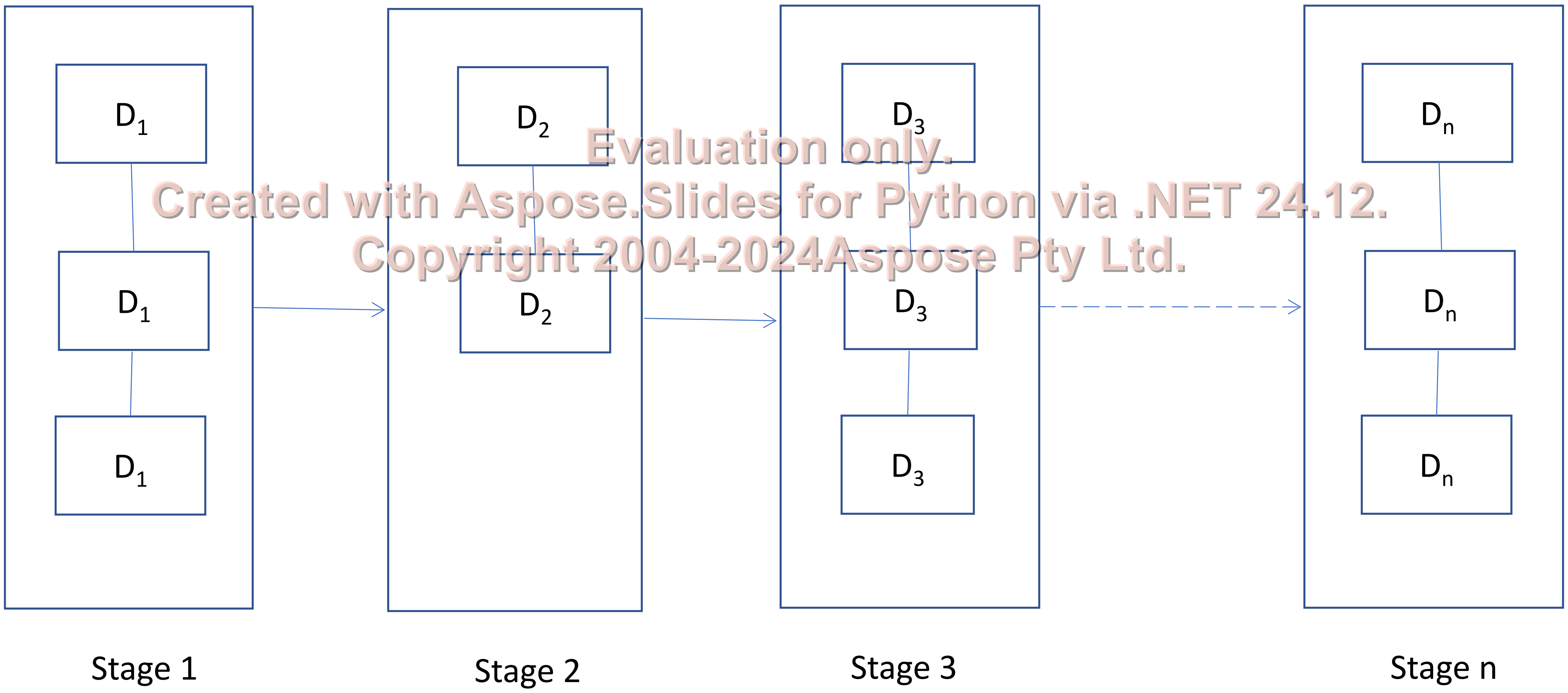


Reliability Design

- It may happen that even if reliability of the individual devices is good but the reliability of the entire system may not be good.
- Hence to obtain the good performance from entire system we can duplicate individual devices.
- Multiple copies of the same device type are connected in parallel through the switching circuits.
- The switching circuits determine which devices in any group are functioning properly.

Reliability Design

Multiple devices connected in parallel in each stage



Reliability Design

- Let stage i contains m_i copies of device D_i.
- The probability that all m_i copies have a malfunction will be given by
- $(1 - r_i)^{m_i}$
- Hence the reliability of stage i will be
- $1 - (1 - r_i)^{m_i}$
- Let the reliability of stage i is denoted by $\phi_i(m_i)$
- $\emptyset_i(m_i) = 1 - (1 - r_i)^{m_i}$
- Here the problem is to use device duplication to maximize the reliability.
- The maximization has to be carried out under a cost constraint.

Reliability Design

- Let c_i be the cost of each unit of device i.
- Let C be the maximum allowable cost of the system to be designed.
- Maximize $\prod_{1 \le i \le n} \emptyset_i\ (m_i)$

- $\sum_{1 \le i \le n} c_i\ m_i\ \le C$

- The upper bound u_i on the cost for the device d_i can be determined as follows.

$$u_i = \left(C + c_i - \sum_{j=1}^{n} c_j \right) \Big/ c_i$$

Reliability Design

- The solution vector S^i for the reliability design problem using dynamic programming consists of the ordered tuples (r, c) i.e. (reliability, cost).
- Dominance Rule
- The ordered tuple (f1, x1) dominates (f2, x2) iff $f1 > f2$ and $x1 < x2$. Then the dominated tuple can be discarded from S^i.

Reliability Design

- Example
- Design a three stage system with device types D_1, D_2, D_3. the cost of the devices are \$30, \$15 and \$20 respectively. The cost of the system is not to be more than \$105. the reliability of each device type is 0.9, 0.8 and 0.5 respectively.
- Solution
- $C = \$105$
- $n = 3$
- $c_1 = \$30$, $c_2 = \$15$, $c_3 = \$20$

Reliability Design

- Computation of u_i

$$u_i = \left(C + c_i - \sum_{j=1}^{n} c_j \right) \Big/ c_i$$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	
3	20	0.5	

- For $i = 1$
- $u_1 = \lfloor (C + c_1 - (c_1 + c_2 + c_3))/c_1 \rfloor$
- $\quad = \lfloor (105 + 30 - (30 + 15 + 20)) / 30 \rfloor$
- $\quad = \lfloor (135 - 65) / 30 \rfloor$
- $\quad = \lfloor (70) / 30 \rfloor$
- $\quad = \lfloor 2.33 \rfloor$
- $u_1 = 2$

Reliability Design

- For i = 2
- $u_2 = \lfloor (C + c_2 – (c_1 + c_2 + c_3))/c_2 \rfloor$
- $= \lfloor (105 + 15 – (30 + 15 + 20)) / 15 \rfloor$
- $= \lfloor (120 – 65) / 15 \rfloor$
- $= \lfloor (55) / 15 \rfloor$
- $= \lfloor 3.66 \rfloor$
- $u_2 = 3$
- For i = 3
- $u_3 = \lfloor (C + c_3 – (c_1 + c_2 + c_3))/c_3 \rfloor$
- $= \lfloor (105 + 20 – (30 + 15 + 20) / 20 \rfloor$
- $= \lfloor (125 – 65) / 20 \rfloor$
- $= \lfloor (60) / 20 \rfloor$
- $= \lfloor 3 \rfloor$
- $u_3 = 3$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Reliability Design

- Initially $S^0 = \{(1, 0)\}$
- Now we will compute $S_j^{\,i}$
- Where i indicates the stage and j indicates number of devices in stage i
- Let us consider device 1 in stage 1
- For Single copy of Device 1 the solution vector will be given by
- $S_1^{\,1} = \{(0.9, 30)\}$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^{\,1} = \{(0.9, 30)\}$
Stage2	

Reliability Design

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

- Now let us consider second copy of device1 in stage 1.
- The reliability of stage 1 with 2 copies of device 1 will be given by

$$\emptyset_i(m_i) = 1 - (1 - r_i)^{m_i}$$

Stages	Solution Vector
Stage1	$S_1^{\,1} = \{(0.9, 30)\}$
Stage2	

- Here i = 1, $m_i = j = 2$, $r_1 = 0.9$
- $\phi_1 (2) = 1- (1 - 0.9)^2$

$= 1 - (0.1)^2$

$= 1 - 0.01$

$\phi_1 (2) = 0.99$

Reliability Design

- The cost of stage 1 with two copies of device 1 will be given by $c_i * m_i$
- $c_i * m_i = c_1 * m_1 = 30 * 2 = 60$
- The solution vector with two copies of device1 in stage 1 will be given by
- $S_2^1 = \{(0.99, 60)\}$
- The solution vector for stage1 will be obtained by combining the ordered tuples S_1^1 and S_2^1
- Therefore $S^1 = \{(0.9, 30), (0.99, 60)\}$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	

Reliability Design

- Let us consider device 2 in stage 2
- For Single copy of Device 2 the solution vector will be given by $S_1^{\,2}$
- With one copy of device2 the ordered pair (r, c) will be (r_2, c_2) i.e. (0.8, 15)
- The first ordered pair in $S_1^{\,2}$ is computed as follows.
- $\{(r_1 * r_2, c_1 + c_2)\} = \{(0.9*0.8, 30+15)\}$
  $= (0.72, 45)$
-

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^{\,1} = \{(0.9, 30)\}$
	$S_2^{\,1} = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^{\,2} = \{(0.72, 45)\}$

Reliability Design

- The second ordered pair in $S_1^{\,2}$ is computed as follows.
- $\{(r_1 * r_2, c_1 + c_2)\} = \{(0.99*0.8, 60+15)\}$ $= (0.792, 75)$
- Therefore the solution vector $S_1^{\,2}$ will consists of the ordered pair
- $S_1^{\,2} = \{(0.72, 45), (0.792, 75)\}$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^{\,1} = \{(0.9, 30)\}$
	$S_2^{\,1} = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^{\,2} = \{(0.72, 45), (0.792, 75)\}$

Reliability Design

- Now let us consider second copy of device2 in stage 2.
- The reliability of stage 2 with 2 copies of device 2 will be given by

$$\phi_i(m_i) = 1 - (1 - r_i)^{m_i}$$

- Here $i = 2$, $m_i = j = 2$, $r_2 = 0.8$
- $\phi_2(2) = 1 - (1 - 0.8)^2$

$= 1 - (0.2)^2$

$= 1 - 0.04$

$\phi_2(2) = 0.96$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 45)\}$

Reliability Design

- Now we will compute S_2^2 .
- The ordered tuples of S_2^2 will be obtained from the ordered tuples of the solution vector S^1 .
- The first ordered tuple will be obtained from (0.9, 30) as follows

$\{(r_1 * \phi_2 (2), c_1 + 2*c_2)\}$

$= \{(0.9*0.96, 30+2*15)\}$

$= (0.864, 60)$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 45)\}$
	$S_2^2 = \{(0.864, 60)\}$

Reliability Design

- The Second ordered tuple will be obtained from (0.99, 60) as follows

$\{(\phi_1 (2) * \phi_2 (2), 2*c_1 + 2*c_2)\}$

$= \{(0.99*0.96, 60+2*15)\}$

$= (0.9504, 90)$

- In the ordered pair (0.9504, 90) the cost is 90.
- The amount left over is 105-90 = 15.
- With the left over amount 15 we can not be able to get device3.
- Therefore the ordered pair (0.9504, 90) will be discarded.

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^{\,1} = \{(0.9, 30)\}$
	$S_2^{\,1} = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^{\,2} = \{(0.72, 45), (0.792, 75)\}$
	$S_2^{\,2} = \{(0.864, 60), \cancel{(0.9504, 90)}\}$

Reliability Design

- Now let us consider third copy of device2 in stage 2.
- The reliability of stage 2 with 3 copies of device 2 will be given by

$$\phi_i(m_i) = 1 - (1 - r_i)^{m_i}$$

- Here i = 2, $m_i = j = 3$, $r_2 = 0.8$
- $\phi_2(3) = 1 - (1 - 0.8)^3$

$$= 1 - (0.2)^3$$

$$= 1 - 0.008$$

$\phi_2(3) = 0.992$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$

Reliability Design

- Now we will compute S_3^2 .
- The ordered tuples of S_3^2 will be obtained from the ordered tuples of the solution vector S^1 .
- The first ordered tuple will be obtained from (0.9, 30) as follows

$\{(r_1 * \phi_2 (3), c_1 + 3*c_2)\}$

$= \{(0.9*0.992, 30+3*15)\}$

$= (0.8928, 75)$

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75)\}$

Reliability Design

- The next ordered tuple in $S_3^{\,2}$ will be obtained from (0.99, 60).

$\{(\phi_1(2) * \phi_2(3), 2*c_1 + 3*c_2)\}$

$= \{(0.99*0.992, 2*30+3*15)\}$

$= (0.98208, 105)$

- In the ordered pair (0.98208, 105) the cost is 105 which is equal to the total system cost.
- We can not be able to procure device3 for the system design.
- Therefore the ordered pair (0.98208, 105) will be discarded.

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^{\,1} = \{(0.9, 30)\}$
	$S_2^{\,1} = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^{\,2} = \{(0.72, 45), (0.792, 75)\}$
	$S_2^{\,2} = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^{\,2} = (0.8928, 75), \cancel{(0.98208, 105)}\}$

Reliability Design

- The ordered tuples obtained in S^2 will be
- $S^2 = \{(0.72, 45), (0.792, 75), (0.864, 60), (0.8928, 75)\}$
- In the above solution vector consider the ordered pairs (0.792, 75), (0.864, 60).
- In the above two ordered tuples the reliability is increasing but the cost is decreasing.
- Therefore by the dominance rule the ordered pair with higher cost will be discarded.
- Therefore the ordered pair (0.792, 75) will be discarded.

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$

Reliability Design

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

- Let us consider device 3 in stage 3
- For Single copy of Device 3 the solution vector will be given by $S_1^{\,3}$
- With one copy of device3 the ordered pair (r, c) will be (r_3, c_3) i.e. (0.5, 20)
- The first ordered pair in $S_1^{\,3}$ is computed from the first ordered pair of S^2.
- $\{(0.72 * r_3, 45 + c_3)\}$
- $= \{(0.72*0.5, 45+20)\}$
- $= (0.36, 65)$

Stages	Solution Vector
Stage1	$S_1^{\,1} = \{(0.9, 30)\}$
	$S_2^{\,1} = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^{\,2} = \{(0.72, 45), (0.792, 75)\}$
	$S_2^{\,2} = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^{\,2} = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^{\,3} = \{(0.36, 65)\}$

Reliability Design

- The next ordered pair in S_1^3 is computed from the ordered pair (0.864, 60) of S^2.
- $\{(0.864 * r_3, 60 + c_3)\} = \{(0.864*0.5, 60+20)\}$
- = (0.432, 80)
-

- The next ordered pair in S_1^3 will be computed from the ordered pair (0.8928, 75) of S^2.
- $\{(0.8928 * r_3, 75 + c_3)\} = \{(0.8928*0.5, 75+20)\}$
- = (0.4464, 95)

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$

Reliability Design

D_i	c_i	r_i	u_i
1	30	0.9	2
2	15	0.8	3
3	20	0.5	3

- Now let us consider second copy of device3 in stage 3.
- The reliability of stage 3 with 2 copies of device3 will be given by

$$\emptyset_i(m_i) = 1 - (1 - r_i)^{m_i}$$

- Here i = 3, $m_i = j = 2$, $r_3 = 0.5$
- $\phi_3\ (2) = 1 - (1 - 0.5)^2$

$$= 1 - (0.5)^2$$

$$= 1 - 0.25$$

$$\phi_3\ (2) = 0.75$$

Stages	Solution Vector
Stage1	$S_1^{\ 1}$ = {(0.9, 30)}
	$S_2^{\ 1}$ = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	$S_1^{\ 2}$ = {(0.72, 45), (0.792, 75)}
	$S_2^{\ 2}$= {(0.864, 60), ~~(0.9504, 90)~~}
	$S_3^{\ 2}$= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	$S_1^{\ 3}$ = {(0.36, 65), (0.432, 80), (0.4464, 95)}

Reliability Design

- With two copy of device3 the ordered pair (r, c) will be (ϕ_3 (2), $2*c_3$) i.e. (0.75, 40)
- Now we will compute S_2^3 .
- The ordered tuples of S_2^3 will be obtained from the ordered tuples of the solution vector S^2 .

•The first ordered pair in S_2^3 is computed from the ordered pair (0.72, 45) of S^2.

$\{(0.72 * \phi_3 (2), 45 + 2*c_3)\}$

$= \{(0.72*0.75, 45+2*20)\}$

$= (0.54, 85)$

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$
	$S_2^3 = \{(0.54, 85)\}$

Reliability Design

- The next ordered pair in S_2^3 is computed from the ordered pair (0.864, 60) of S^2.

$\{(0.864 * \phi_3 (2), 60 + 2*c_3)\}$
$= \{(0.864*0.75, 60+2*20)\}$
$= (0.648, 100)$

- The next ordered pair in S_2^3 is computed from the ordered pair (0.8928, 75) of S^2.

$\{(0.8928 * \phi_3 (2), 75 + 2*c_3)\}$
$= \{(0.8928*0.75, 75+2*20)\}$
$= (0.6696, 115)$

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$
	$S_2^3 = \{(0.54, 85), (0.648, 100), (0.6696, 115)\}$

Reliability Design

- In the solution vector S_2^3 the ordered pair (0.6696, 115) is having the cost 115.
- Since the cost is exceeding the available amount the ordered pair (0.6696, 115) will be discarded from S_2^3.

Stages	Solution Vector
Stage1	S_1^1 = {(0.9, 30)}
	S_2^1 = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	S_1^2 = {(0.72, 45), (0.792, 75)}
	S_2^2= {(0.864, 60), ~~(0.9504, 90)~~}
	S_3^2= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	S_1^3 = {(0.36, 65), (0.432, 80), (0.4464, 95)}
	S_2^3 = {(0.54, 85), (0.648, 100), ~~(0.6696, 115)~~}
	S_3^3 =

Reliability Design

- Now let us consider third copy of device3 in stage 3.
- The reliability of stage 3 with 3 copies of device 3 will be given by

$$\emptyset_i(m_i) = 1 - (1 - r_i)^{m_i}$$

- Here i = 3, m_i = j = 3, r_3 = 0.5
- ϕ_3 (3) = 1- (1 – 0.5)^3

  = 1 – $(0.5)^3$

  = 1 – 0.125

ϕ_2 (3) = 0.875

Stages	Solution Vector
Stage1	$S_1^{\,1}$ = {(0.9, 30)}
	$S_2^{\,1}$ = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	$S_1^{\,2}$ = {(0.72, 45), (0.792, 75)}
	$S_2^{\,2}$= {(0.864, 60), ~~(0.9504, 90)~~}
	$S_3^{\,2}$= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	$S_1^{\,3}$ = {(0.36, 65), (0.432, 80), (0.4464, 95)}
	$S_2^{\,3}$ = {(0.54, 85), (0.648, 100), ~~(0.6696, 115)~~}
	$S_3^{\,3}$ =

Reliability Design

- With two copy of device3 the ordered pair (r, c) will be (ϕ_3 (3), 2*c_3) i.e. (0.875, 60)
- Now we will compute S_3^3 .
- The ordered tuples of S_3^3 will be obtained from the ordered tuples of the solution vector S^2 .
- The first ordered pair in S_3^3 is computed from the ordered pair (0.72, 45) of S^2.

$\{(0.72 * \phi_3 (3), 45 + 3*c_3)\}$

$= \{(0.72*0.875, 45+3*20)\}$

$= (0.63, 105)$

Stages	Solution Vector
Stage1	$S_1^{\ 1} = \{(0.9, 30)\}$
	$S_2^{\ 1} = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^{\ 2} = \{(0.72, 45), (0.792, 75)\}$
	$S_2^{\ 2} = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^{\ 2} = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^{\ 3} = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$
	$S_2^{\ 3} = \{(0.54, 85), (0.648, 100), \cancel{(0.6696, 115)}\}$
	$S_3^{\ 3} = \{(0.63, 105)\}$

Reliability Design

- The next ordered pair in S_3^3 is computed from the ordered pair (0.864, 60) of S^2.

$\{(0.864 * \phi_3 (3), 60 + 3*c_3)\}$
$= \{(0.864*0.875, 60+3*20)\}$
$= (0.756, 120)$

- The next ordered pair in S_3^3 is computed from the ordered pair (0.8928, 75) of S^2.

$\{(0.8928 * \phi_3 (3), 75 + 3*c_3)\}$
$= \{(0.8928*0.875, 75+3*20)\}$
$= (0.7812, 135)$

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$
	$S_2^3 = \{(0.54, 85), (0.648, 100), \cancel{(0.6696, 115)}\}$
	$S_3^3 = \{(0.63, 105). (0.756, 120), (0.7812, 135)\}$

Reliability Design

- In the solution vector S_3^3 the ordered pairs (0.756, 120) and (0.7812, 135) will be discarded because the cost exceeds the available amount.
- The solution vector of stage 3 will be given by $S^3 = \{S_1^3, S_2^3, S_3^3\}$
- Hence S^3 = {(0.36, 65), (0.432, 80), (0.4464, 95), (0.54, 85), (0.63, 105), (0.648, 100)}

Stages	Solution Vector
Stage1	S_1^1 = {(0.9, 30)}
	S_2^1 = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	S_1^2 = {(0.72, 45), (0.792, 75)}
	S_2^2= {(0.864, 60), ~~(0.9504, 90)~~}
	S_3^2= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	S_1^3 = {(0.36, 65), (0.432, 80), (0.4464, 95)}
	S_2^3 = {(0.54, 85), (0.648, 100), ~~(0.6696, 115)~~}
	S_3^3 = {(0.63, 105), ~~(0.756, 120), (0.7812, 135)~~}
	S^3 = {(0.36, 65), (0.432, 80), (0.4464, 95), (0.54, 85), (0.63, 105), (0.648, 100)}

Reliability Design

- In the solution vector S^3 consider the ordered pairs (0.4464, 95) and (0.54, 85).
- In the above two ordered tuples the reliability is increasing but the cost is decreasing.
- Therefore by dominance rule the ordered tuple with higher cost i. e. (0.4464, 95) will be discarded.

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$
	$S_2^3 = \{(0.54, 85), (0.648, 100), \cancel{(0.6696, 115)}\}$
	$S_3^3 = \{(0.63, 105), \cancel{(0.756, 120), (0.7812, 135)}\}$
	$S^3 = \{(0.36, 65), (0.432, 80), \cancel{(0.4464, 95)}, (0.54, 85), (0.63, 105), (0.648, 100)\}$

Reliability Design

- In the solution vector S^3 consider the ordered pairs (0.63, 105) and (0.648, 100).
- In the above two ordered tuples the reliability is increasing but the cost is decreasing.
- Therefore by dominance rule the ordered tuple with higher cost i. e. (0.63, 105) will be discarded.

Stages	Solution Vector
Stage1	S_1^1 = {(0.9, 30)}
	S_2^1 = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	S_1^2 = {(0.72, 45), (0.792, 75)}
	S_2^2= {(0.864, 60), ~~(0.9504, 90)~~}
	S_3^2= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	S_1^3 = {(0.36, 65), (0.432, 80), (0.4464, 95)}
	S_2^3 = {(0.54, 85), (0.648, 100), ~~(0.6696, 115)~~}
	S_3^3 = {(0.63, 105), ~~(0.756, 120), (0.7812, 135)~~}
	S^3 = {(0.36, 65), (0.432, 80), ~~(0.4464, 95)~~, (0.54, 85), ~~(0.63, 105)~~, (0.648, 100)}
	S^3 = {(0.36, 65), (0.432, 80), (0.54, 85), (0.648, 100)}

Reliability Design

- In the solution vector S^3 the ordered pairs (0.648, 100) is having the maximum reliability.
- The above ordered tuple has been obtained from S_2^3.
- For the solution vector S_2^3
- $i = 3$, $j = 2$ and $m_3 = 2$
- Therefore the number of copies of device type3 in stage 3 is 2.

Stages	Solution Vector
Stage1	S_1^1 = {(0.9, 30)}
	S_2^1 = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	S_1^2 = {(0.72, 45), (0.792, 75)}
	S_2^2= {(0.864, 60), ~~(0.9504, 90)~~}
	S_3^2= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	S_1^3 = {(0.36, 65), (0.432, 80), (0.4464, 95)}
	S_2^3 = {(0.54, 85), (0.648, 100), ~~(0.6696, 115)~~}
	S_3^3 = {(0.63, 105), ~~(0.756, 120), (0.7812, 135)~~}
	S^3 = {(0.36, 65), (0.432, 80), ~~(0.4464, 95)~~, (0.54, 85), ~~(0.63, 105)~~, (0.648, 100)}
	S^3 = {(0.36, 65), (0.432, 80), (0.54, 85), (0.648, 100)}

Reliability Design

- The ordered pairs (0.648, 100) has been computed from (0.864, 60).
- The ordered tuple (0.864, 60) has been obtained from S_2^2.
- For the solution vector S_2^2
- $i = 2$, $j = 2$ and $m_2 = 2$
- Therefore the number of copies of device type2 in stage 2 is 2.

Stages	Solution Vector
Stage1	$S_1^1 = \{(0.9, 30)\}$
	$S_2^1 = \{(0.99, 60)\}$
	$S^1 = \{(0.9, 30), (0.99, 60)\}$
Stage2	$S_1^2 = \{(0.72, 45), (0.792, 75)\}$
	$S_2^2 = \{(0.864, 60), \cancel{(0.9504, 90)}\}$
	$S_3^2 = \{(0.8928, 75), \cancel{(0.98208, 105)}\}$
	$S^2 = \{(0.72, 45), \cancel{(0.792, 75)}, (0.864, 60), (0.8928, 75)\}$
	$S^2 = \{(0.72, 45), (0.864, 60), (0.8928, 75)\}$
Stage3	$S_1^3 = \{(0.36, 65), (0.432, 80), (0.4464, 95)\}$
	$S_2^3 = \{(0.54, 85), (0.648, 100), \cancel{(0.6696, 115)}\}$
	$S_3^3 = \{(0.63, 105), \cancel{(0.756, 120), (0.7812, 135)}\}$
	$S^3 = \{(0.36, 65), (0.432, 80), \cancel{(0.4464, 95)}, (0.54, 85), \cancel{(0.63, 105)}, (0.648, 100)\}$
	$S^3 = \{(0.36, 65), (0.432, 80), (0.54, 85), (0.648, 100)\}$

Reliability Design

- The ordered pairs (0.864, 60) has been computed from (0.9, 30).
- The ordered tuple (0.9, 30) has been obtained from S_1^1 .
- For the solution vector S_1^1
- $i = 1, j = 1$ and $m_1 = 1$
- Therefore the number of copies of device type1 in stage 1 is 1.

Stages	Solution Vector
Stage1	S_1^1 = {(0.9, 30)}
	S_2^1 = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	S_1^2 = {(0.72, 45), (0.792, 75)}
	S_2^2= {(0.864, 60), ~~(0.9504, 90)~~}
	S_3^2= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	S_1^3 = {(0.36, 65), (0.432, 80), (0.4464, 95)}
	S_2^3 = {(0.54, 85), (0.648, 100), ~~(0.6696, 115)~~}
	S_3^3 = {(0.63, 105), ~~(0.756, 120), (0.7812, 135)~~}
	S^3 = {(0.36, 65), (0.432, 80), ~~(0.4464, 95)~~, (0.54, 85), ~~(0.63, 105)~~, (0.648, 100)}
	S^3 = {(0.36, 65), (0.432, 80), (0.54, 85), (0.648, 100)}

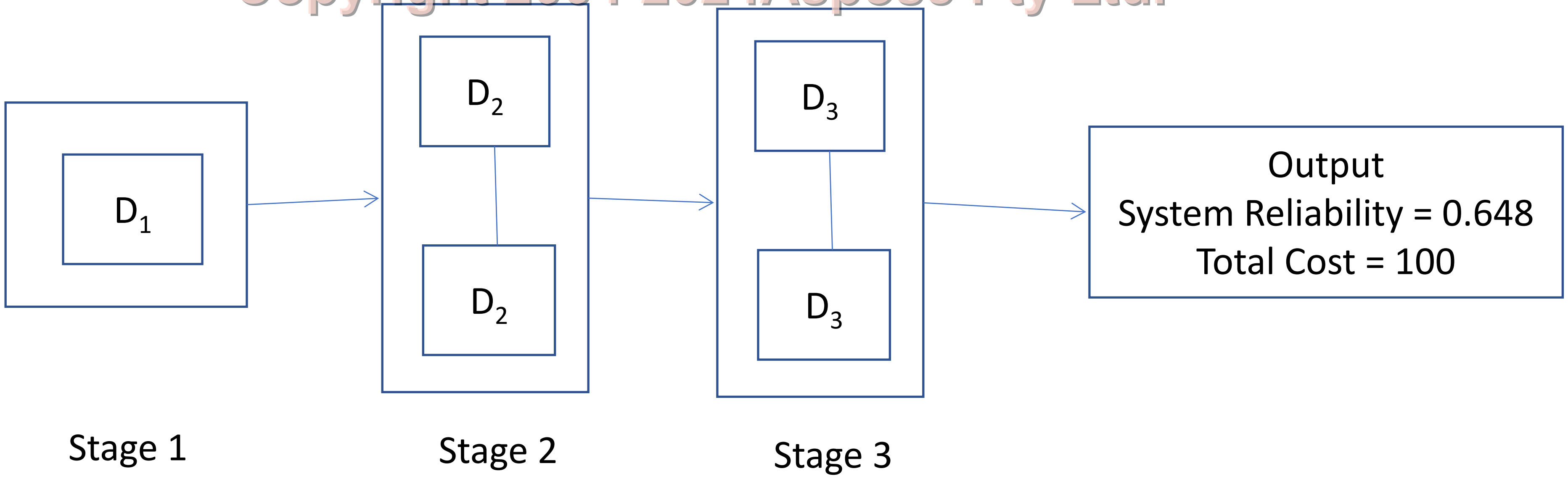
Reliability Design

- Therefore the 3stage system will consists of following device types.
- No. of copies device type1 = m1 =1
- No. of copies device type2 = m2 = 2
- No. of copies device type3 = m3 = 2

Stages	Solution Vector
Stage1	S_1^1 = {(0.9, 30)}
	S_2^1 = {(0.99, 60)}
	S^1 = {(0.9, 30), (0.99, 60)}
Stage2	S_1^2 = {(0.72, 45), (0.792, 75)}
	S_2^2= {(0.864, 60), ~~(0.9504, 90)~~}
	S_3^2= {(0.8928, 75), ~~(0.98208, 105)~~}
	S^2 = {(0.72, 45), ~~(0.792, 75)~~, (0.864, 60), (0.8928, 75)}
	S^2 = {(0.72, 45), (0.864, 60), (0.8928, 75)}
Stage3	S_1^3 = {(0.36, 65), (0.432, 80), (0.4464, 95)}
	S_2^3 = {(0.54, 85), (0.648, 100), ~~(0.6696, 115)~~}
	S_3^3 = {(0.63, 105), ~~(0.756, 120), (0.7812, 135)~~}
	S^3 = {(0.36, 65), (0.432, 80), ~~(0.4464, 95)~~, (0.54, 85), ~~(0.63, 105)~~, (0.648, 100)}
	S^3 = {(0.36, 65), (0.432, 80), (0.54, 85), (0.648, 100)}

Reliability Design

- Therefore the 3stage system will consists of following device types.
- No. of copies device type1 = 1
- No. of copies device type2 = 2
- No. of copies device type3 = 2

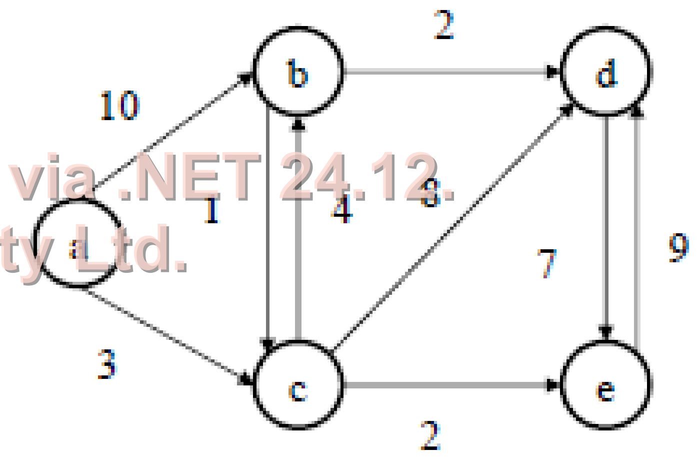


All Pair Shortest Path

- Let G =(V, E) be a directed graph.
- The all pairs shortest path problem is to determine the shortest path between every pair of vertices in a directed graph G.
- That is, for every pair of vertices (i, j), we have to find a shortest path from i to j as well as one from j to i.

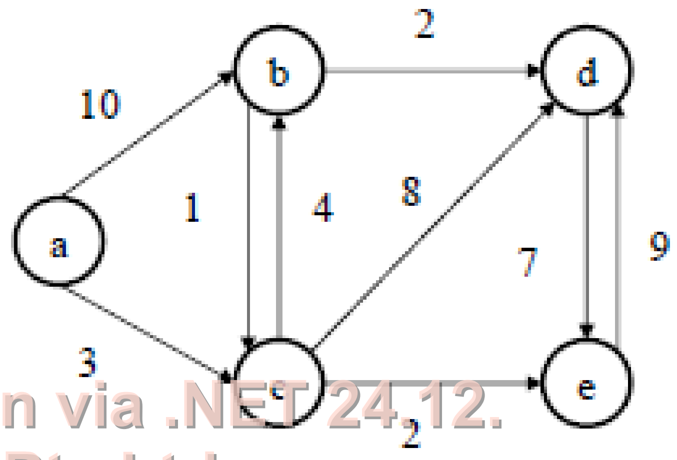
All Pair Shortest Path

- Consider the following Graph.
- Objective: Find out the shortest path for each pair of vertices.
- Shortest Path by Considering a as Source
- a to b
- a to c
- a to d
- a to e
- Shortest Path by Considering b as Source
- b to a
- b to c
- b to d
- b to e



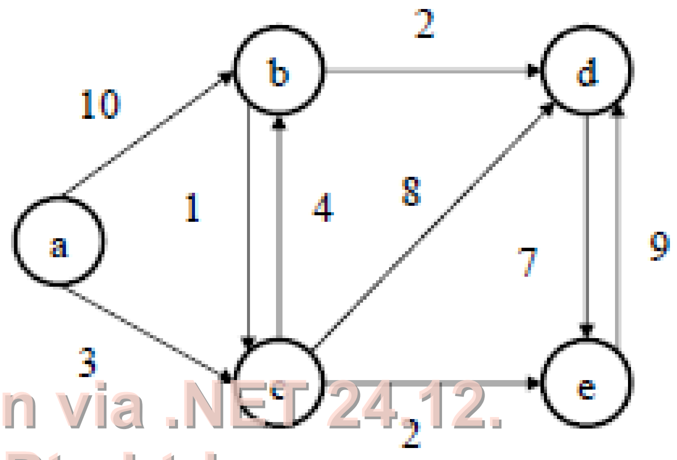
All Pair Shortest Path

- Shortest Path by Considering c as Source
- c to a
- c to b
- c to d
- c to e
- Shortest Path by Considering d as Source
- d to a
- d to b
- d to c
- d to e



All Pair Shortest Path

- Shortest Path by Considering e as Source
- e to a
- e to b
- e to c
- e to d



All Pair Shortest Path

- Let *cost* be the cost adjacency matrix for G such that
- cost[i, i] = 0, 1<= i <= n
- cost [i, j] = weight of the edge (i, j), if(i, j) ϵ E(G)
- cost [i, j] = ∞, if i ≠ j and (i, j) ! ϵ E(G)

All Pair Shortest Path

- The shortest i to j path in G, $i \neq j$ originates at vertex i and goes through some intermediate vertices (possibly none) and terminates at vertex j.
- If k is an intermediate vertex on this shortest path, then the subpaths from i to k and from k to j must be shortest paths from i to k and k to j, respectively.
- Otherwise, the i to j path is not of minimum length.
- Therefore the principle of optimality holds.
- Let $A^k(i, j)$ represent the length of a shortest path from i to j going through no vertex of index greater than k, we obtain:

$$A^k(i,j) = \min\ \{A^{k-1}(i,j),\ A^{k-1}(i,k) + A^{k-1}(k,j)\},\quad k \geq 1$$

All Pair Shortest Path

- Algorithm All Paths (cost, A, n)
- // cost [1:n, 1:n] is the cost adjacency matrix of a graph with n vertices;
- //A [i, j] is the cost of a shortest path from vertex i to vertex j.
- //cost [i, i] = 0.0 for 1 < i < n
- {
- for i := 1 to n
- do for j:= 1 to n
- do A^0 [i, j] := cost [i, j]; // copy cost into A.
- for k := 1 to n
- do for i := 1 to n
- do for j := 1 to n
- do A^k [i, j] := min {A^{k-1} [i, j], (A^{k-1} [i, k] + A^{k-1} [k, j])};
- }

All Pair Shortest Path

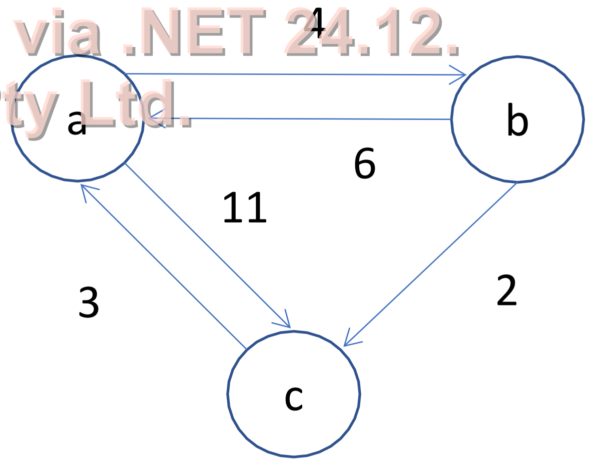
$$A(i,j) = \min \; \{ \min_{1 \le k \le n} \{A^{k-1}(i,k) + A^{k-1}(k,j)\}, cost(i,j)\}$$

$$A^k(i,j) = \min \; \{A^{k-1}(i,j), A^{k-1}(i,k) + A^{k-1}(k,j)\}, \quad k \ge 1$$

All Pair Shortest Path

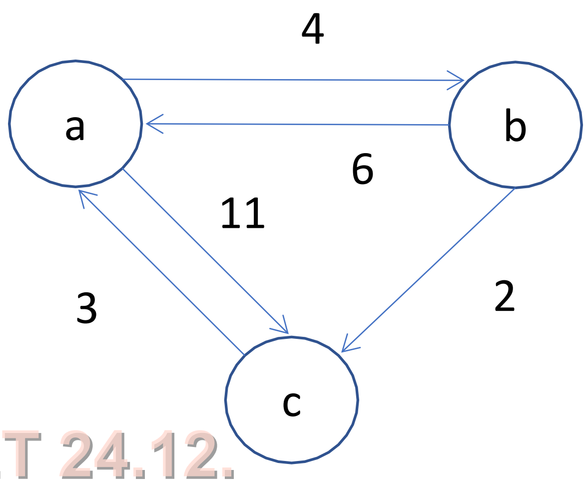
- Example
- Compute all pair shortest path for the following graph using Floyd's algorithm.
- Solution
- For the given graph n = 3
- The cost matrix for the given graph is

$$\cos t = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix}$$



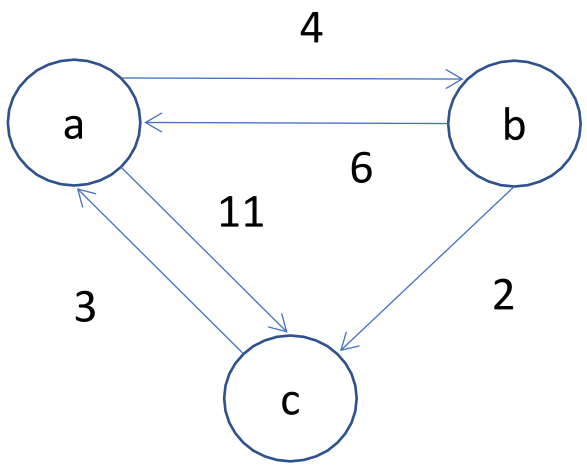
All Pair Shortest Path

- Copy the cost matrix to A^0
- for $i = 1$ to 3
- for $j = 1$ to 3
- $A^0 [i, j] = cost [i, j]$



$$A^0 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix}$$

All Pair Shortest Path

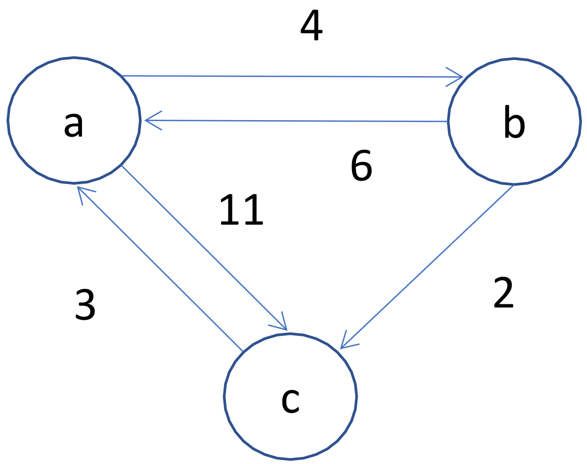


- for k := 1 to 3
- for i := 1 to 3
- for j := 1 to 3
- k= 1, i = 1, j = 1
- $A^1[1, 1] = \min \{(A^0[1, 1] + A^0[1, 1]), A^0[1, 1]\} = \min \{0 + 0, 0\} = 0$
- k= 1, i = 1, j = 2
- $A^1[1, 2] = \min \{(A^0[1, 1] + A^0[1, 2]), A^0[1, 2]\} = \min \{(0 + 4), 4\} = 4$
- k= 1, i = 1, j = 3
- $A^1[1, 3] = \min \{(A^0[1, 1] + A^0[1, 3]), A^0[1, 3]\} = \min \{(0 + 11), 11\} = 11$

$$A^0 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix}$$

$$A^1 = \begin{bmatrix} 0 & 4 & 11 \\ & & \\ & & \end{bmatrix}$$

All Pair Shortest Path

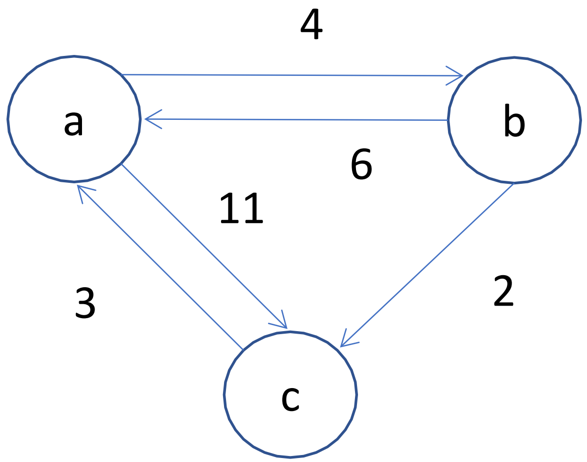


- $k = 1, i = 2, j = 1$
- $A^1[2, 1] = \min\{(A^0[2, 1] + A^0[1, 1]), A^0[2, 1]\} = \min\{(6 + 0), 6\} = 6$
- $k = 1, i = 2, j = 2$
- $A^1[2, 2] = \min\{(A^0[2, 1] + A^0[1, 2]), A^0[2, 2]\} = \min\{(6 + 4), 0)\} = 0$
- $k = 1, i = 2, j = 3$
- $A^1[2, 3] = \min\{(A^0[2, 1] + A^0[1, 3]), A^0[2, 3]\} = \min\{(6 + 11), 2\} = 2$
- $k = 1, i = 3, j = 1$
- $A^1[3, 1] = \min\{(A^0[3, 1] + A^0[1, 1]), A^0[3, 1]\} = \min\{(3 + 0), 3\} = 3$
- $k = 1, i = 3, j = 2$
- $A^1[3, 2] = \min\{(A^0[3, 1] + A^0[1, 2]), A^0[3, 2]\} = \min\{(3 + 4), \infty\} = 7$
- $k = 1, i = 3, j = 3$
- $A^1[3, 3] = \min\{(A^0[3, 1] + A^0[1, 3]), A^0[3, 3]\} = \min\{(3 + 11), 0\} = 0$

$$A^0 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix}$$

$$A^1 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

All Pair Shortest Path

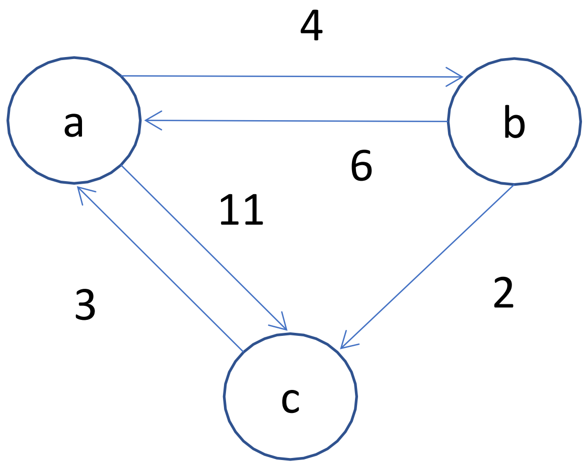


- for k := 1 to 3
- for i := 1 to 3
- for j := 1 to 3
- k= 2, i = 1, j = 1
- $A^2[1, 1] = \min\{(A^1[1, 2] + A^1[2, 1]), A^1[1, 1]\} = \min\{4 + 6, 0\} = 0$
- k= 2, i = 1, j = 2
- $A^2[1, 2] = \min\{(A^1[1, 2] + A^1[2, 2]), A^1[1, 2]\} = \min\{(4 + 0), 4\} = 4$
- k= 2, i = 1, j = 3
- $A^2[1, 3] = \min\{(A^1[1, 2] + A^1[2, 3]), A^1[1, 3]\} = \min\{(4 + 2), 11\} = 6$

$$A^1 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ & & \\ & & \end{bmatrix}$$

All Pair Shortest Path

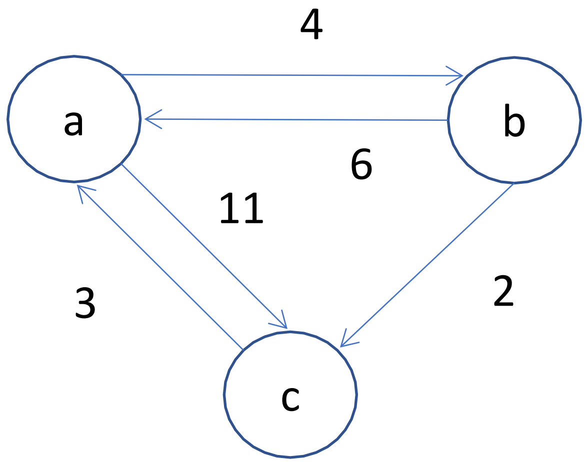


- k= 2, i = 2, j = 1
- $A^2[2, 1] = \min\{(A^1[2, 2] + A^1[2, 1]), A^1[2, 1]\} = \min\{(0 + 6), 6\} = 6$
- k= 2, i = 2, j = 2
- $A^2[2, 2] = \min\{(A^1[2, 2] + A^1[2, 2]), A^1[2, 2]\} = \min\{(0 + 0), 0)\} = 0$
- k= 2, i = 2, j = 3
- $A^2[2, 3] = \min\{(A^1[2, 2] + A^1[2, 3]), A^1[2,3]\} = \min\{(0 + 2), 2\} = 2$
- k= 2, i = 3, j = 1
- $A^2[3, 1] = \min\{(A^1[3, 2] + A^1[2, 1]), A^1[3, 1]\} = \min\{(7 + 6), 3\} = 3$
- k= 2, i = 3, j = 2
- $A^2[3, 2] = \min\{(A^1[3, 2] + A^1[2, 2]), A^1[3, 2]\} = \min\{(7 + 0), 7\} = 7$
- k= 2, i = 3, j = 3
- $A^2[3, 3] = \min\{(A^1[3, 2] + A^1[2, 3]), A^1[3, 3]\} = \min\{(7 + 2), 0\} = 0$

$$A^1 = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

All Pair Shortest Path

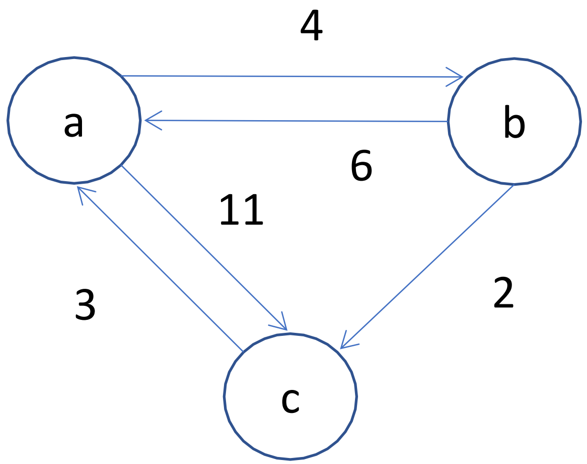


- for k := 1 to 3
- for i := 1 to 3
- for j := 1 to 3
- k= 3, i = 1, j = 1
- $A^3 [1, 1] = \min \{(A^2 [1, 3] + A^2 [3, 1]), A^2 [1, 1]\} = \min \{6 + 3, 0\} = 0$
- k= 3, i = 1, j = 2
- $A^3 [1, 2] = \min \{(A^2 [1, 3] + A^2 [3, 2]), A^2 [1, 2]\} = \min \{(6 + 7), 4\} = 4$
- k= 3, i = 1, j = 3
- $A^3 [1, 3] = \min \{(A^2 [1, 3] + A^2 [3, 3]), A^2 [1, 3]\} = \min \{(6+ 0), 6\} = 6$

$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 0 & 4 & 6 \\ & & \\ & & \end{bmatrix}$$

All Pair Shortest Path



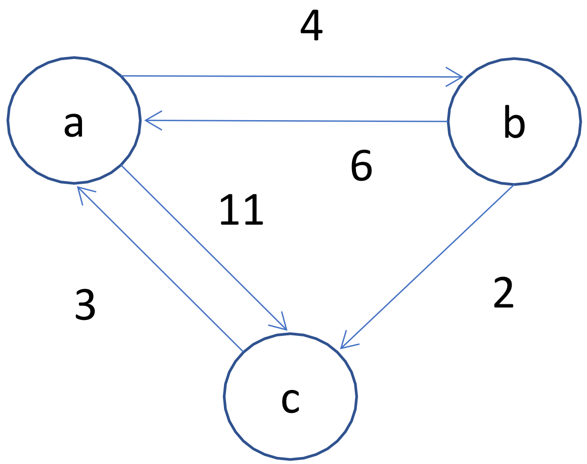
- k= 3, i = 2, j = 1
- $A^3[2, 1] = \min\{(A^2[2, 3] + A^2[3, 1]), A^2[2, 1]\} = \min\{(2 + 3), 6\} = 5$
- k= 3, i = 2, j = 2
- $A^3[2, 2] = \min\{(A^2[2, 3] + A^2[3, 2]), A^2[2, 2]\} = \min\{(2 + 7), 0)\} = 0$
- k= 3, i = 2, j = 3
- $A^3[2, 3] = \min\{(A^2[2, 3] + A^2[3, 3]), A^2[2, 3]\} = \min\{(2 + 0), 2\} = 2$
- k= 3, i = 3, j = 1
- $A^3[3, 1] = \min\{(A^2[3, 3] + A^2[3, 1]), A^2[3, 1]\} = \min\{(0 + 3), 3\} = 3$
- k= 3, i = 3, j = 2
- $A^3[3, 2] = \min\{(A^2[3, 3] + A^2[3, 2]), A^2[3, 2]\} = \min\{(0 + 7), 7\} = 7$
- k= 3, i = 3, j = 3
- $A^3[3, 3] = \min\{(A^2[3, 3] + A^2[3, 3]), A^2[3, 3]\} = \min\{(0 + 0), 0\} = 0$

$$A^2 = \begin{bmatrix} 0 & 4 & 6 \\ 6 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$$A^3 = \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

All Pair Shortest Path

- k = 4 False
- The Algorithm will return A^3
- The shortest path for each pair of vertices are



Vertex Pair	Path cost / length
a → b	4
a → c	6
b → a	5
b → c	2
c → a	3
c → b	7

$$A^3 = \begin{bmatrix} 0 & 4 & 6 \\ 5 & 0 & 2 \\ 3 & 7 & 0 \end{bmatrix}$$

$$\cos t = \begin{bmatrix} 0 & 4 & 11 \\ 6 & 0 & 2 \\ 3 & \infty & 0 \end{bmatrix}$$

Travelling Salesperson Problem

- Let $G = (V, E)$ be a directed graph with edge costs c_{ij}.
- The variable c_{ij} is defined such that $c_{ij} > 0$ for all i and j and $c_{ij} = \infty$ if $< i, j> \, !\epsilon \, E$.
- Let $|V| = n$ and assume $n > 1$.
- A tour of G is a directed simple cycle that includes every vertex in V.
- The cost of a tour is the sum of the cost of the edges on the tour.
- The travelling sales person problem is to find a tour of minimum cost.
- The tour is to be a simple path that starts and ends at vertex 1.

Travelling Salesperson Problem

- Every tour consists of an edge (1, k) for some k ϵ V – {1} and a path from vertex k to vertex 1.
- The path from vertex k to vertex 1 goes through each vertex in the set V – {1, k} exactly once.
- If the tour is optimal then the path from k to 1 must be shortest.
- Therefore the principle of optimality holds.
- Let g(i, S) be the length of a shortest path starting at vertex i going through all the vertices in S and terminates at vertex i.

Travelling Salesperson Problem

- The function g(1, V – {1}) is the length of an optimal salesperson tour.
- From the principle of optimality we can have

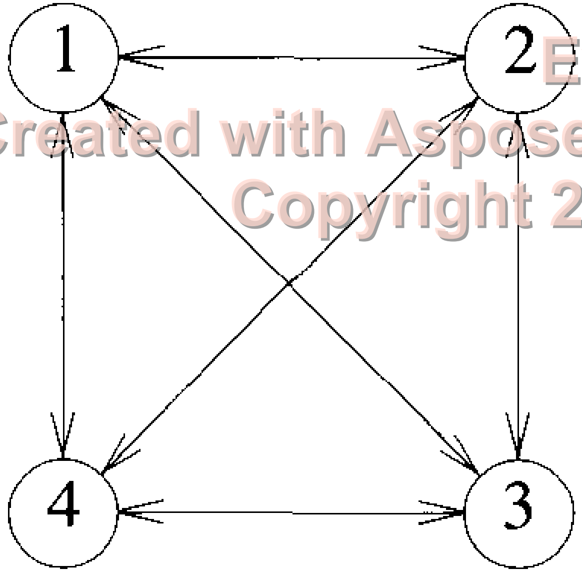
$$g(1, V - \{1\}) = \min_{2 \le k \le n}\{c_{ik} + g(k, V - \{1, k\})\}$$

- Generalizing the above equation we obtain

$$g(i, S) = \min_{j \in S}\{c_{ij} + g(j, S - \{j\})\}$$

Travelling Salesperson Problem

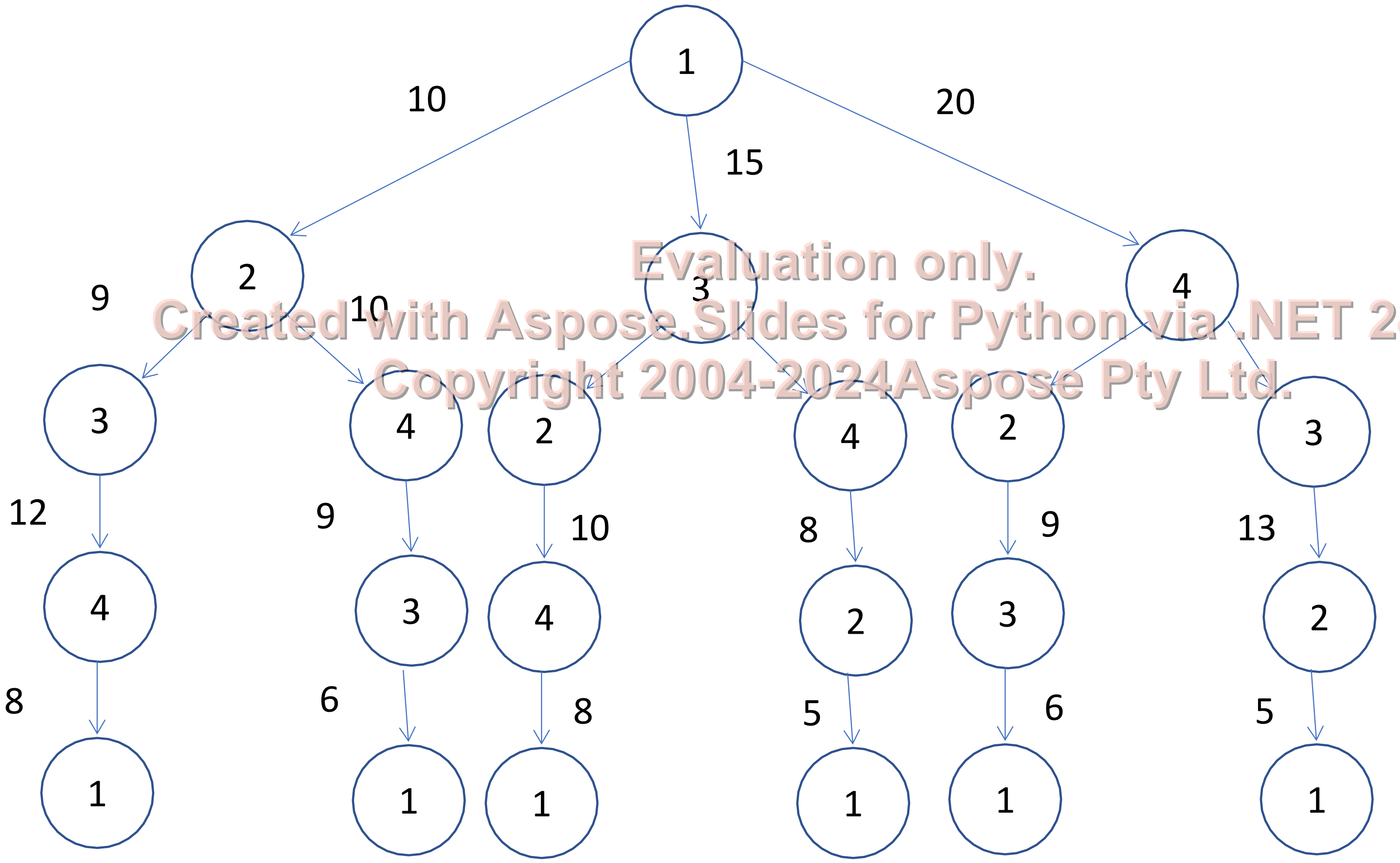
- Let us consider the following diagraph and the corresponding cost adjacency matrix.

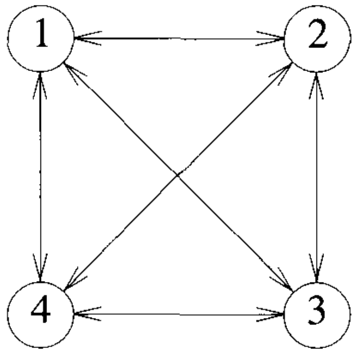


$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

- For the above diagraph we need to find out the optimal cost tour.

Travelling Salesperson Problem





$$\begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

Travelling Salesperson Problem

- The optimal cost tour for the salesperson will be given by

$$g(1,\{2,3,4\}) = \min_{j \in S}\{c_{12} + g(2,\{3,4\}), c_{13} + g(3,\{2,4\}), c_{14} + g(4,\{2,3\})\}$$

$$g(2,\{3,4\}) = \min_{j \in S}\{c_{23} + g(3,\{4\}), c_{24} + g(4,\{3\})\}$$

$$g(3,\{2,4\}) = \min_{j \in S}\{c_{32} + g(2,\{4\}), c_{34} + g(4,\{2\})\}$$

$$g(4,\{2,3\}) = \min_{j \in S}\{c_{42} + g(2,\{3\}), c_{43} + g(3,\{2\})\}$$

Travelling Salesperson Problem

$$g(3,\{4\}) = \min_{j \in S}\{c_{34} + g(4,\{\phi\})\}$$

$$g(4,\{3\}) = \min_{j \in S}\{c_{43} + g(3,\{\phi\})\}$$

$$g(2,\{4\}) = \min_{j \in S}\{c_{24} + g(4,\{\phi\})\}$$

$$g(4,\{2\}) = \min_{j \in S}\{c_{42} + g(2,\{\phi\})\}$$

$$g(2,\{3\}) = \min_{j \in S}\{c_{23} + g(3,\{\phi\})\}$$

$$g(3,\{2\}) = \min_{j \in S}\{c_{32} + g(2,\{\phi\})\}$$

$$g(2,\{\phi\}) = c_{21}$$

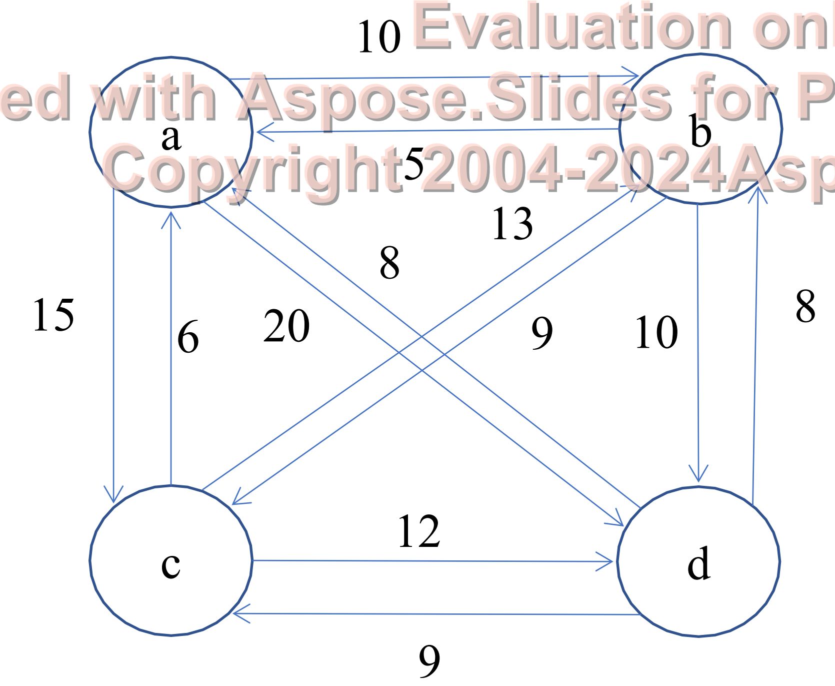
$$g(3,\{\phi\}) = c_{31}$$

$$g(4,\{\phi\}) = c_{41}$$

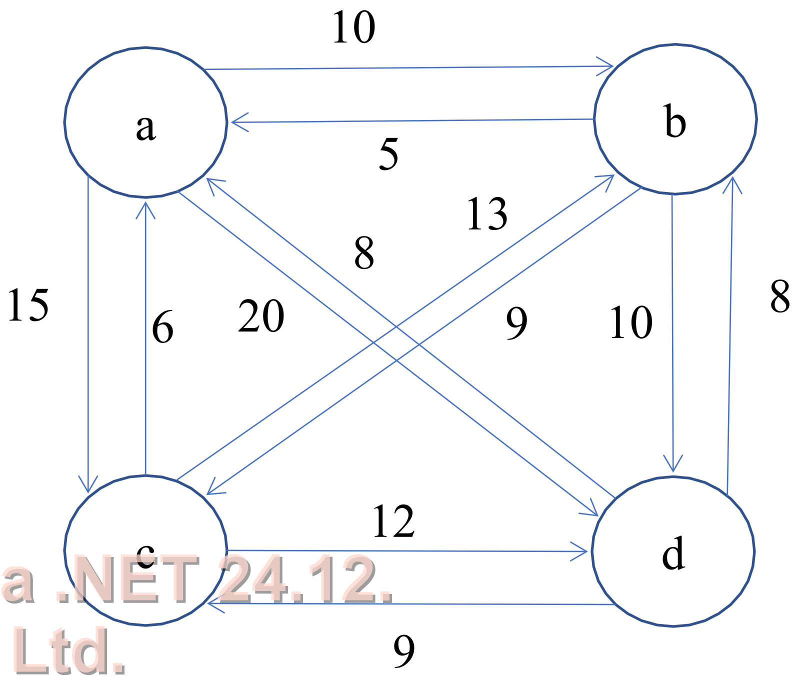
- The optimal tour will be obtained using the bottom up approach by using the above functions.

Travelling Salesperson Problem

- Example
- For the given diagraph obtain the optimum cost tour for travelling sales person.



Travelling Salesperson Problem



- Solution
- The cost adjacency matrix for the given diagraph is

$$cost = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

- We will select the arbitrary vertex say a as the starting vertex.
- Now we will find the function values g(i, S) for the intermediate sets with increasing size.
- Here the size of the sets will be S = ϕ, 1, 2, 3.

Travelling Salesperson Problem

$$\text{cost} = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

- Step 1: Let us consider $S = \phi$
- The corresponding function values will be as follows
- $g(2, \phi) = c_{21} = 5$
- $g(3, \phi) = c_{31} = 6$
- $g(4, \phi) = c_{41} = 8$
- Step 2: Let us consider $S = 1$
- The corresponding function values will be as follows
- $g(2, \{3\}) = c_{23} + g(3, \phi) = 9 + 6 = 15$
- $g(2, \{4\}) = c_{24} + g(4, \phi) = 10 + 8 = 18$
- $g(3, \{2\}) = c_{32} + g(2, \phi) = 13 + 5 = 15$

Set Size	Functions	Values
ϕ	$g(2, \phi)$	5
	$g(3, \phi)$	6
	$g(4, \phi)$	8
1	$g(2, \{3\})$	15
	$g(2, \{4\})$	18
	$g(3, \{2\})$	15
	$g(3, \{4\})$	
	$g(4, \{2\})$	
	$g(4, \{3\})$	

Travelling Salesperson Problem

$$cost = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

- $g(3, \{4\}) = c_{34} + g(4, \phi) = 12 + 8 = 20$
- $g(4, \{2\}) = c_{42} + g(2, \phi) = 8 + 5 = 13$
- $g(4, \{3\}) = c_{43} + g(3, \phi) = 9 + 6 = 15$
- Step 3: Let us consider S = 2
- The corresponding function values will be as follows
- $g(2, \{3, 4\}) = \min \{c_{23} + g(3, \{4\}), c_{24} + g(4, \{3\})\}$
- $= \min \{ 9 + 20, 10 + 15\}$
- $= \min \{ 29, 25\}$
- $= 25$

Set Size	Functions	Values
ϕ	$g(2, \phi)$	5
	$g(3, \phi)$	6
	$g(4, \phi)$	8
1	$g(2, \{3\})$	15
	$g(2, \{4\})$	18
	$g(3, \{2\})$	15
	$g(3, \{4\})$	20
	$g(4, \{2\})$	13
	$g(4, \{3\})$	15
2	$g(2, \{3, 4\})$	25

Travelling Salesperson Problem

$$\cos t = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

- $g(3, \{2, 4\}) = \min \{c_{32} + g(2, \{4\}), c_{34} + g(4, \{2\})\}$
- $= \min \{ 13 + 18, 12 + 13\}$
- $= \min \{ 31, 25\}$
- $= 25$

- $g(4, \{2, 3\}) = \min \{c_{42} + g(2, \{3\}), c_{43} + g(3, \{2\})\}$
- $= \min \{ 8 + 15, 9 + 15\}$
- $= \min \{ 23, 24\}$
- $= 23$

Set Size	Functions	Values
ϕ	$g(2, \phi)$	5
	$g(3, \phi)$	6
	$g(4, \phi)$	8
1	$g(2, \{3\})$	15
	$g(2, \{4\})$	18
	$g(3, \{2\})$	15
	$g(3, \{4\})$	20
	$g(4, \{2\})$	13
	$g(4, \{3\})$	15
2	$g(2, \{3, 4\})$	25
	$g(3, \{2, 4\})$	25
	$g(4, \{2, 3\})$	23

Travelling Salesperson Problem

$$\text{cost} = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

- Step4: Let us consider S = 3
- The corresponding function values will be as follows
- g(1, {2, 3, 4}) = min {c_{12} + g(2, {3,4}), c_{13} + g(3, {2,4}), c_{14} + g(4, {2,3}) }
- Substituting the function values from the table we have
- g(1, {2, 3, 4}) = min { 10 + 25, 15 + 25, 20 + 23 }
- = min { 35, 40, 43 }
- = 35
- The optimum tour cost is 35.

Set Size	Functions	Values
ϕ	g(2, ϕ)	5
	g(3, ϕ)	6
	g(4, ϕ)	8
1	g(2, {3})	15
	g(2, {4})	18
	g(3, {2})	15
	g(3, {4})	20
	g(4, {2})	13
	g(4, {3})	15
2	g(2, {3, 4})	25
	g(3, {2, 4})	25
	g(4, {2, 3})	23

Travelling Salesperson Problem

$$cost = \begin{bmatrix} 0 & 10 & 15 & 20 \\ 5 & 0 & 9 & 10 \\ 6 & 13 & 0 & 12 \\ 8 & 8 & 9 & 0 \end{bmatrix}$$

- Consider Step 4 the minimum value has been obtained from c_{12} + g(2, {3,4}) i. e. 35
- In step 3 the minimum value for the function g(2, {3,4}) has been obtained from c_{24} + g(4, {3}) i. e. 25
- In step 2 the minimum value for the function g(4, {3}) has been obtained from c_{43} + g(3, ϕ) i. e. 15.
- In step 1 the minimum value for the function g(3, ϕ) has been obtained from c_{31} i. e. 6
- 1 → 2 → 4 → 3 → 1
- The optimum tour cost is 35.

Set Size	Functions	Values
ϕ	g(2, ϕ)	5
	g(3, ϕ)	6
	g(4, ϕ)	8
1	g(2, {3})	15
	g(2, {4})	18
	g(3, {2})	15
	g(3, {4})	20
	g(4, {2})	13
	g(4, {3})	15
2	g(2, {3, 4})	25
	g(3, {2, 4})	25
	g(4, {2, 3})	23