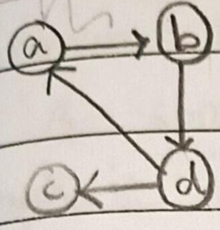
Warshall's algorithm:



$$R^{(0)} = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 0 & 0 \\ b & 0 & 0 & 0 & 1 \\ c & 0 & 0 & 0 & 0 \\ d & 1 & 0 & 1 & 0 \end{array}$$

and

or

non trivial = 1

$0 \times 1 = 0 + 0 = 0$

$$R^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

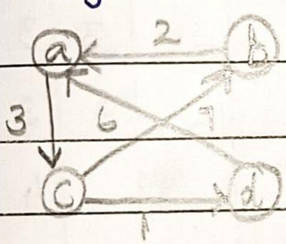
1+1

$$R^{(2)} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$R^{(3)} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

$$R^{(4)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Floyd's algorithm:



$$D^{(0)} = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & \infty & 3 & \infty \\ b & 2 & 0 & \infty & \infty \\ c & \infty & 7 & 0 & 1 \\ d & 6 & \infty & \infty & 0 \end{array}$$

$(2+\infty, 0)$

$$D^{(1)} = \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ \infty & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix}$$

$$D^{(2)} = \begin{bmatrix} 0 & \infty & 3 & \infty \\ 2 & 0 & 5 & \infty \\ 9 & 7 & 0 & 1 \\ 6 & \infty & 9 & 0 \end{bmatrix}$$

$$D^{(3)} = \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 9 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix}$$

$$D^{(4)} = \begin{bmatrix} 0 & 10 & 3 & 4 \\ 2 & 0 & 5 & 6 \\ 7 & 7 & 0 & 1 \\ 6 & 16 & 9 & 0 \end{bmatrix}$$

Optimal BST

no. of trees constructed for given nodes.

Catalon number $= \frac{1}{n+1}\binom{2n}{n}$

$n=4$ $\quad = \frac{1}{5} \times {}^{8}C_{4} = \frac{1}{5} \times \frac{8\times7\times6\times5}{1\times2\times3\times4} = 14$

Key	A	B	C	D
	1	2	3	4
Probability	0.1	0.2	0.4	0.3

Main table

0 to n
1 to n+1
$(i,i-1)=(1,1-1)=1,0=0$
$(i,i) = P_i$

	0	1	2	3	4
1	0	0.1	0.4	1.1	1.7
2		0	0.2	0.8	1.4
3			0	0.4	1.0
4				0	0.3
5					0

Root table

	0	1	2	3	4
1		1	2	3	3
2			2	3	3
3				3	3
4					4
5					

(3,4)

(3,4) 1) $0+0.3=0.3+(0.4+0.3)=1.0$
$0.4+0=0.4+0.7=1.1$ } min

(2,3) 2) $0+0.3=0.3+0.6=0.9$
$0.2+0=0.2+0.6=0.8$ ✓

(2,4)

(2,4) 3) $0+1.0=1.0+0.9=1.9$
$Pr = 2+3+4$
$0.2+0.3=0.5+0.9=1.4$
$0.8+0=0.8+0.9=1.7$

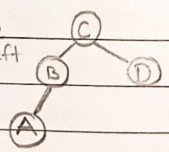
(1,2) $0+0.2=0.2+0.3=0.5$
$0+0.1=0.1+0.3=0.4$

(1,3) $0+0.8=0.8+0.7=1.5$
1+2+3
$0.1+0.4=0.5+0.7=1.2$
$0.4+0=0.4+0.7=1.1$

(1,4)
$1.4+1.0=2.4$
$1.1+1.0=2.1$
$0.7+1.0=1.7$
$1.1+1.0=2.1$

(1,4) = 3 (root node)

(1,2) = 2
B will be left child



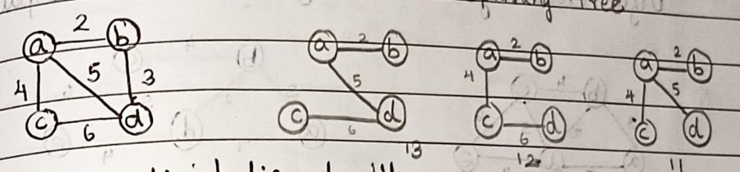
total probability $0.4\times1+0.2\times2+0.3\times2+0.1\times3$
$=0.4+0.4+0.6+0.3=1.7$

Minimum spanning tree:

A spanning tree of an undirected connected graph is it's connected acyclic subgraph (that is tree) that contains all vertices of the graph.

Minimum spanning tree is it's spanning tree of the smallest weight where the weight of the tree is defined as the sum of the weights on all it's edges (it includes all vertices of graph not edges)

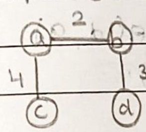
Possible spanning trees



MST (i) Kriskal's algorithm:

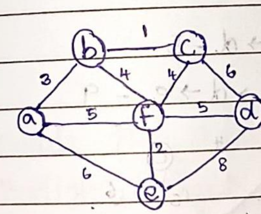
* write the edges in ascending order based on weight
* Then ignore if there any edge form cyclic

① a-b = 2 ② b-d = 3 ③ a-c-4 ④ a-d = 5 ⑤ c-d = 6

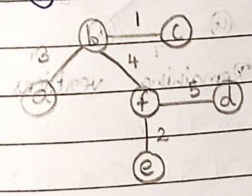


cost = 9

1\.



b-c = 1
f-e = 2
b-a = 3
b-f = 4
c-f = 4 ×
a-f = 5 ×
f-d = 5
c-d = 6 ×
a-e = 6 ×
e-d = 8



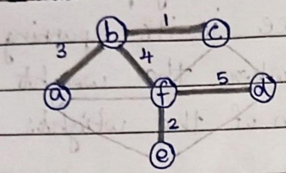
cost = 15

(ii) Prim's algorithm:

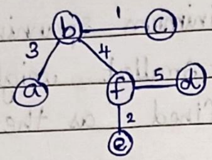
Tree vertices	Remaining vertices:
a(-,-)	min(b(a,3), f(a,5), e(a,6), c(-,∞), d(-,∞))
b(a,3)	c(b,1), f(b,4), e(a,6), d(-,∞)
c(b,1)	f(b,4), d(c,6), e(a,6)

f(b,4)	d(f,5), e(f,2)
e(f,2)	d(f,5)
d(f,5)	-

Illustration:



MST

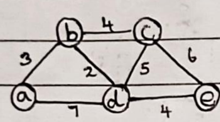


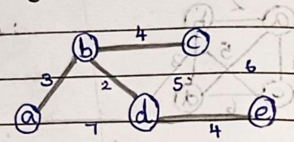
cost = 1

8.5.2023 Dijikstra's algorithm - Single source shortest path algorithm

* based on Prim's algorithm.

①



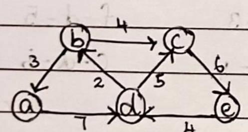


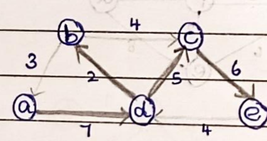
Tree vertices	Remaining vertices
a(-,-)	b(a,3), d(a,7), c(a,∞), e(-,∞)
b(a,3)	c(b,7) [4+3], d(b,5) [2+3], e(-,∞)
d(b,5)	c(b,7), e(d,9)
c(b,7)	e(d,9)
e(d,9)	-

a→b = 3 a→b→d = 5

a→b→c = 7 a→b→d→e = 9

②





Tree vertex	Remaining vertices
(a(-,-))	d(a,7), b(-,∞), c(-,∞), e(-,∞)
d(a,7)	b(d,9), c(d,12), e(-,∞)
b(d,9)	c(d,12), e(-,∞)
c(d,12)	e(c,18)
e(c,18)	

Shortest path from source node a

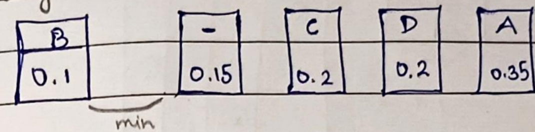
a→d = 7 a→d→c = 12

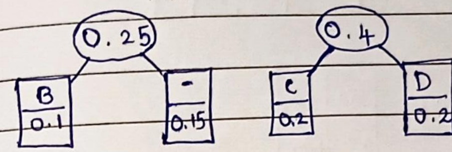
a→d→b - 9 a→d→c→e = 18

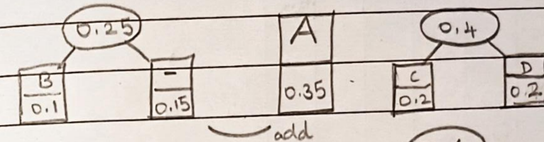
Huffman tree's & codes

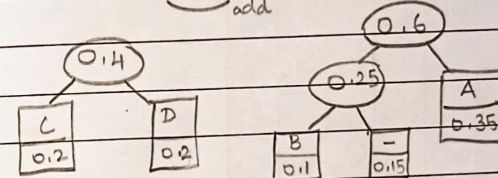
symbol	A	B	C	D	-
frequency	0.35	0.1	0.2	0.2	0.15

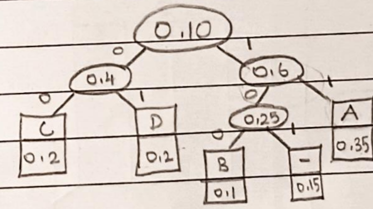
ascending order:











left subtree = 0

right subtree = 1

Encode the text DAD decode the string 100110110 1101

Prefix-free code → A - 11 D - 01

B - 100 - - 101 (code words)

C - 00

text DAD string - 100110110 1101

011101 B A D - AD

Prefix for one symbol is not a code word of another symbol

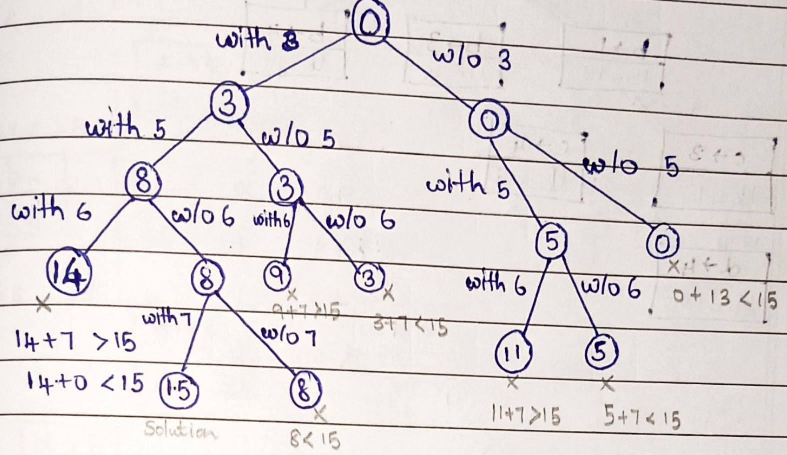
* encoding type - Variable length encoding (huffman)

fixed length encoding (ASCII codeword same)

Given a set $A = \{a_1, a_2, \ldots a_n\}$ subset whose sum is equal to integer d. $A = \{1, 2, 5, 6, 8\}$ $d = 9$

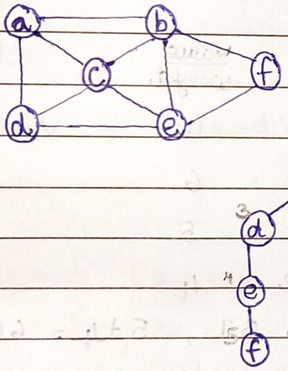
$\{1, 2, 6\}$ $\{8, 1\}$ form a subset for d value

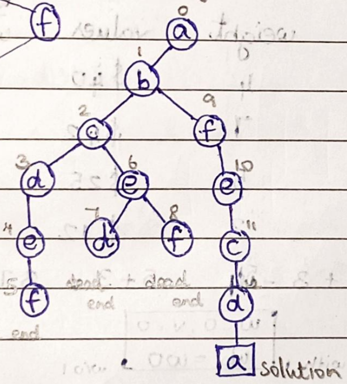
$A = \{3, 5, 6, 7\}$, $d = 15$



Hamiltonian circuit Problem

: Start from source node visit each vertex only one time after visit all nodes atlast visit source node.





Branch and bound technique:

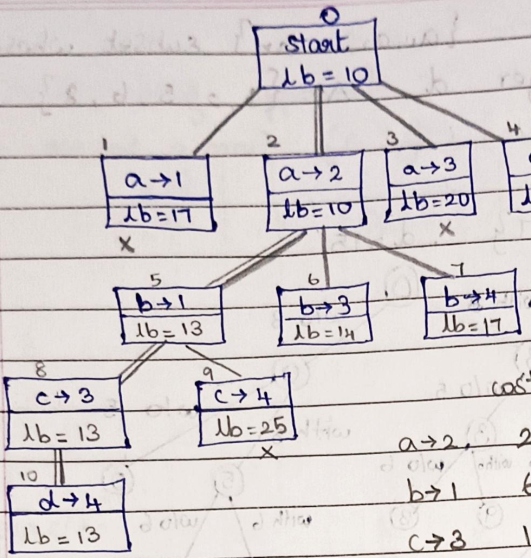
Assignment problem:

	job1	job2	job3	job4	
C =	9	2	7	8	person a
	6	4	3	7	person b
	5	8	1	8	person c
	7	6	9	4	person d

lower bound:

possible min value for each person: $2+3+1+4 = 10$

ans : $2 + 6 + 1 + 4 = 13$ (min spent for each person (only one job assigning for each person)



job 1 → a → lb = 9+3+1+4 = 17

job 2 → a → lb = 2+3+1+4 = 10

b1 = 2+6+1+4 = 13

b3 = 2+3+5+4 = 14

b4 = 2+7+1+7 = 17

c3 = 2+6+1+4 = 13

c4 = 2+6+8+9 = 25

d4 = 2+6+1+4 = 13

	cost
a→2	2
b→1	6
c→3	1
d→4	4
	13

5.5.2023

Knapsack problem:

$ub = v + (W - w)(v_{i+1} / w_{i+1})$

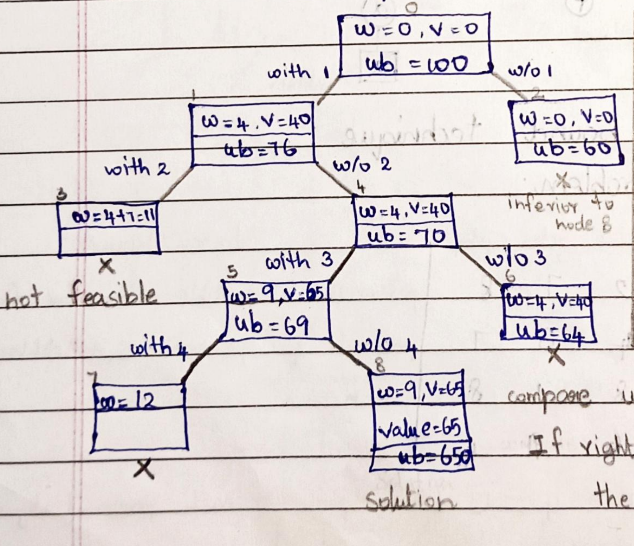
W = capacity of knapsack, v = value, w = weight

item	weight	values	value/weight
1	4	\$40	10
2	7	\$42	6
3	5	\$25	5
4	3	\$12	4

W = 10

ans: w + w = v + v

7 + 3 = 54, 5 + 3 = 37, ✓5 + 4 = 65, 4 + 3 = 52



ub = 0 + (10 - 0) × 10 = 100

ub = 40 + (10 - 4) × 6 = 76

ub = 0 + (10 - 0) × 6 = 60

ub = 40 + (10 - 4)(5) = 70

ub = 65 + (10 - 9)(4) = 69

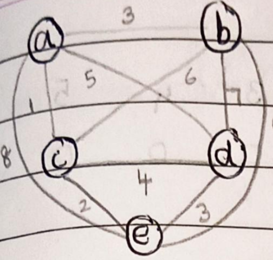
ub = 40 + (10 - 4) 4 = 64

ub = 65 + (10 - 9) × 10 = 650

compare ub 65 > 60 so 65 is solution

If right side ub ≥ left side ub calculate the remaining ub for other items.

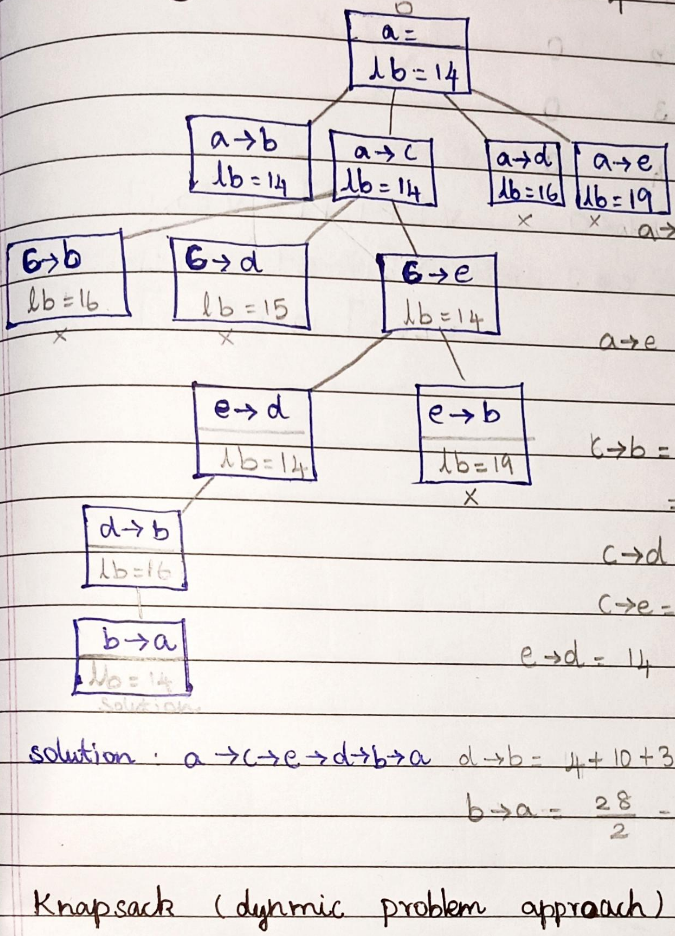
Traveling Salesman Problem :



ans: a → c → e → d → b → a

$lb = [(1+3)+(3+6)+(1+2)+(4+3)+(3+2)/2]$

$lb = 28/2 = 14$



lb =

$a \to d = ((1+5)+(3+6)+(1+2)+(3+5)+(3+2))/2$

$= 16 \qquad = \frac{31}{2} = 15.5$

$a \to e = (1+8)+9+3+7+2+8 \div 2$

$= \frac{38}{2} = 19$

$c \to b = 4+9+1+6+7+5/2$

$= \frac{32}{2} = 16$

$c \to d = 4+9+(1+4)+7+5/2 = \frac{30}{2} = 15$

$c \to e = 14$

$e \to d = 14 \quad e \to b = 4+(3+9)+3+7+(9+2)/2$

$= \frac{37}{2} = 19$

solution : a → c → e → d → b → a $d \to b = 4+10+3+10+5/2 = 32/2 = 16$

$b \to a = \frac{28}{2} = 14$

Knapsack (dynmic problem approach)

→ Problem deviding into overlap sub problem (subproblem aren't independent)

item	weight	value
1	2	12
2	1	10
3	3	20
4	2	15

W = 5 (capacity)

$$V[i,j] = \begin{cases} \max[V[i-1,j], v_i + V[i-1, j-w_i]] & \text{if } j-w_i \geq 0 \\ V[i-1,j] & \text{if } j-w_i < 0 \end{cases}$$