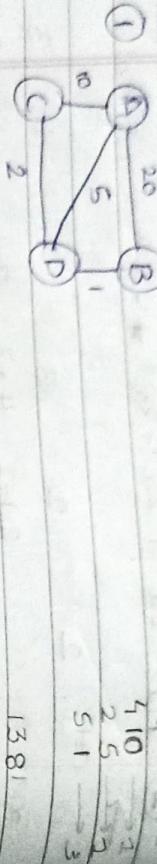


UNIT - 3

GREEDY TECHNIQUE
PRIMS ALGORITHM

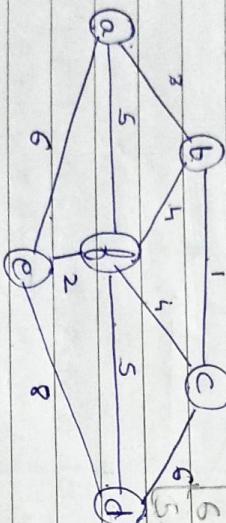


A B-20
C-10
D-5

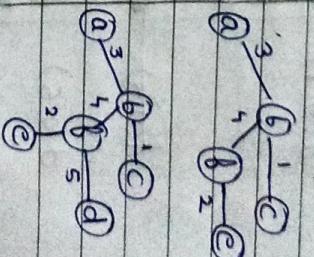
B-1

C-2

(2)



Minimum ~~cost~~ = $1 + 3 + 4 + 5 + 2 = 15$
of Spanning Tree



29.50

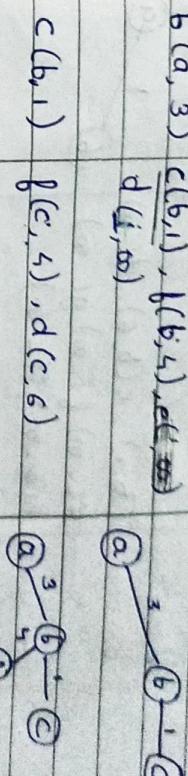
59.20

410

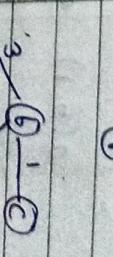
25

51

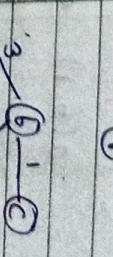
3



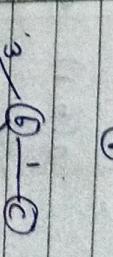
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



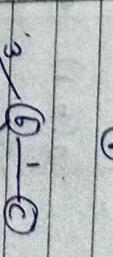
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



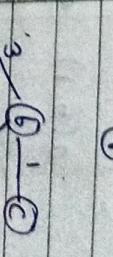
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



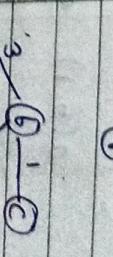
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



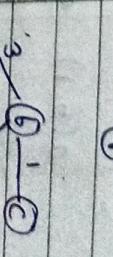
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



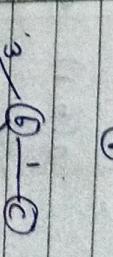
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



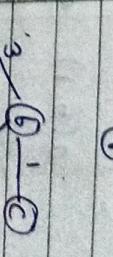
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



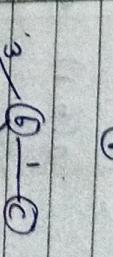
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



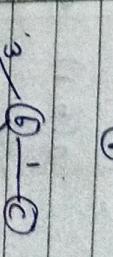
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



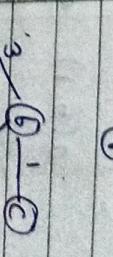
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



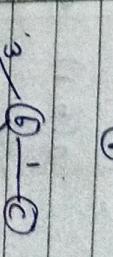
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



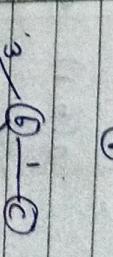
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



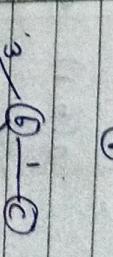
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



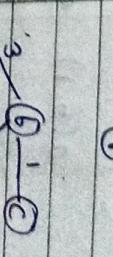
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



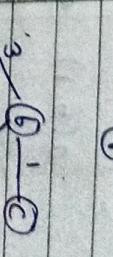
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



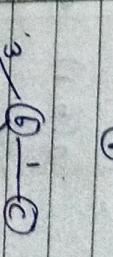
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



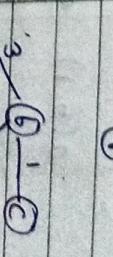
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



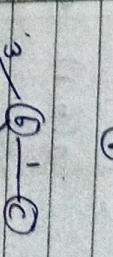
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



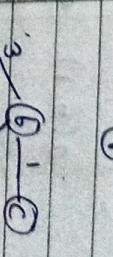
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



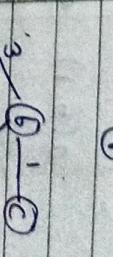
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



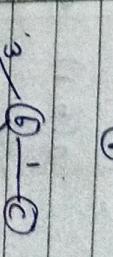
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



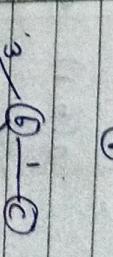
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



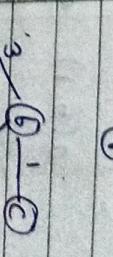
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



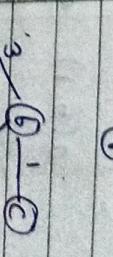
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



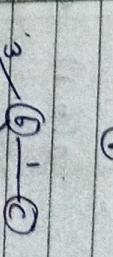
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



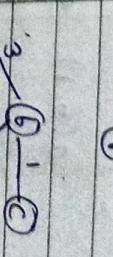
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



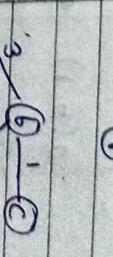
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



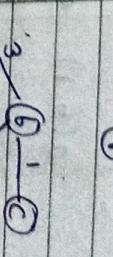
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



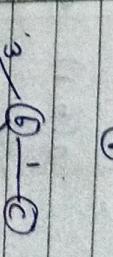
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



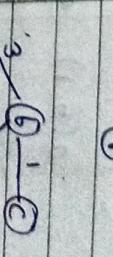
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



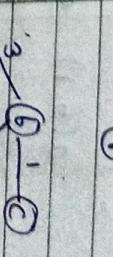
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



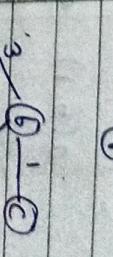
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



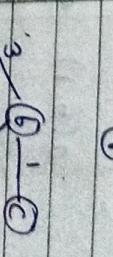
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



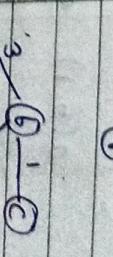
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



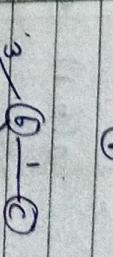
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



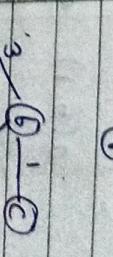
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



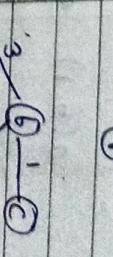
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



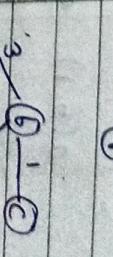
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



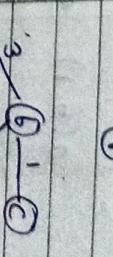
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



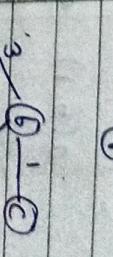
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



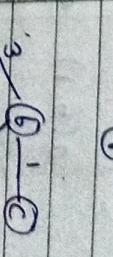
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



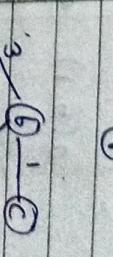
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



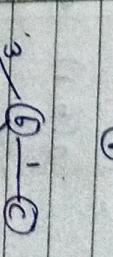
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



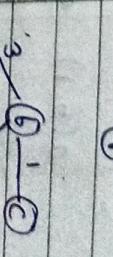
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



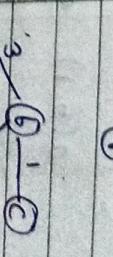
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



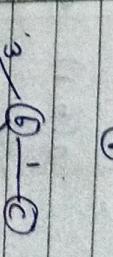
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



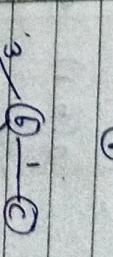
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



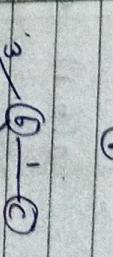
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



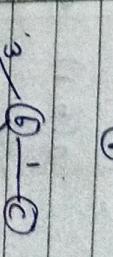
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



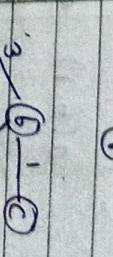
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



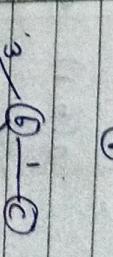
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



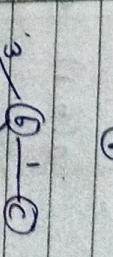
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



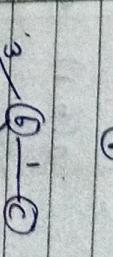
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



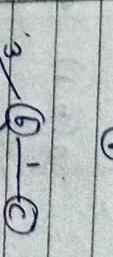
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



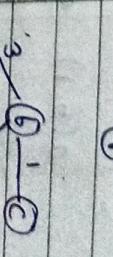
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



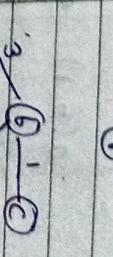
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



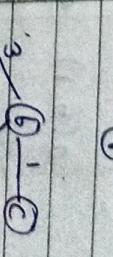
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



~~f(b,1)~~
~~f(c,4)~~, d(c,6)



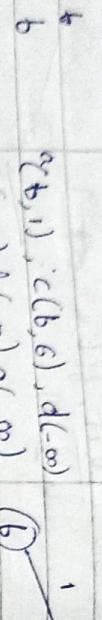
~~f(b,1)~~
~~f(c,4)~~, d(c,6)



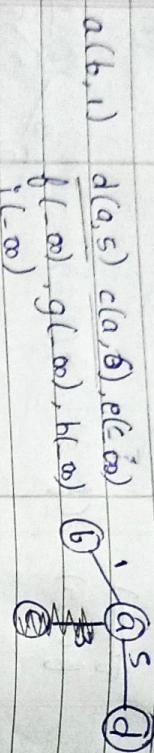
Tree Vertices Remaining Vertices

Illustration

b
 $c(b, 1), c(b, 6), d(-\infty)$

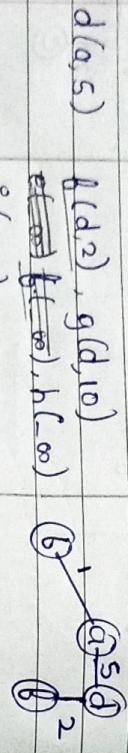


$e(-\infty)$, $f(-\infty)$, $g(-\infty)$, $h(-\infty)$



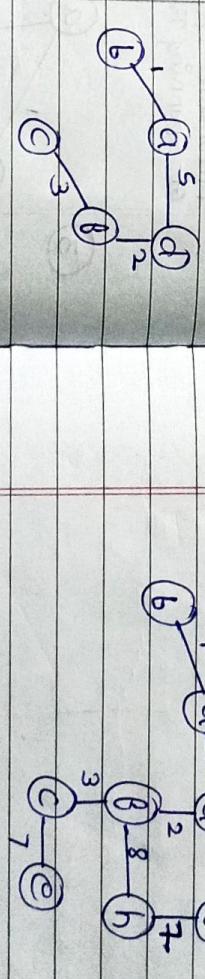
$h(f, 8)$, $g(i, 7)$, $i(h, 8)$

$d(c, 3)$, $e(c, 7)$, $f(-\infty)$

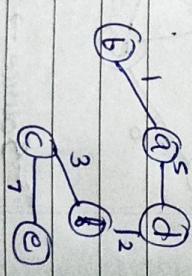


$g(h, 7)$, $i(g, 3)$

$f(d, 2)$
 $c(f, 3), h(f, 8)$
 $e(i(-\infty), e(-\infty))$



$c(b, 3)$, $e(e, 7); i(-\infty)$



$MST = 1 + 5 + 2 + 8 + 3 + 7 + 7 + 3$
 $= 36$

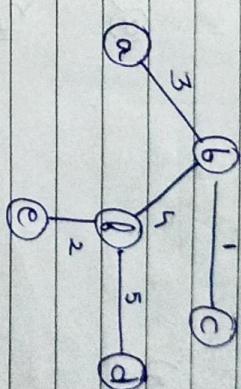
Algorithm Primes(a)

$V_1 \leftarrow \{V_0\}$

$E_1 \leftarrow \emptyset$

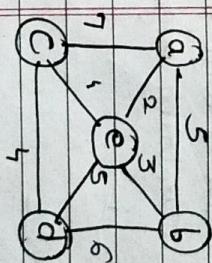
for $i=1$ to b do

end



$$MST = 3 + 1 + 4 + 5 + 2 = 15$$

Primes



$$\textcircled{a} (c, -, +) \quad b(a, 5) \quad \underline{e(a, 2)} \quad \underline{i(a, 1)} \quad \textcircled{a} \quad 2 \\ d(-, \infty) \quad \textcircled{a}$$

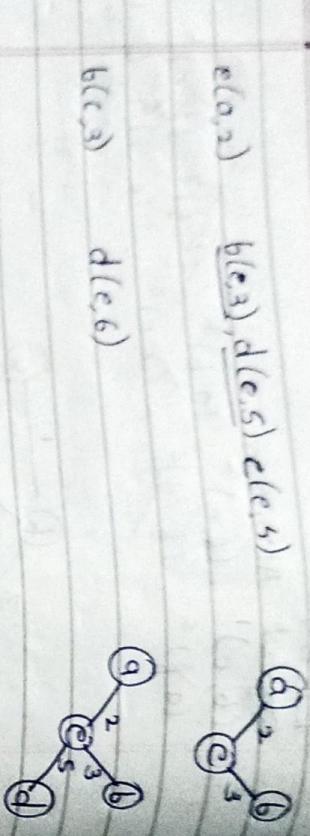
Kruskall

Algorithm

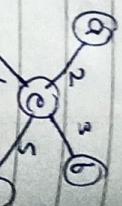
$(b, c)^1, (b, e)^2, (a, b)^3, (b, d)^4, (b, c)^5, (a, f)^5, (b, d)^5, (c, e)^6, (c, d)^6, (c, d)^7$

$e(a,2)$ $b(c,3), d(c,5) \quad e(c,4)$

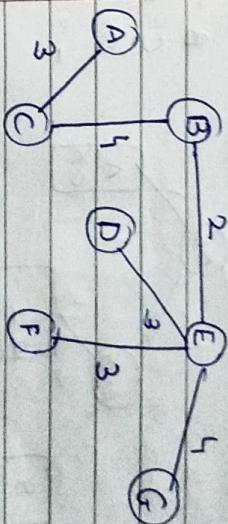
$b(c,3)$ $d(c,6)$



$d(c,5)$ $c(d,4)$



MST: $2+3+4+5 = 14$



MST: $3+1+2+3+3+4 = 19$

Algorithm KRUSKAL'S(G)

- Sort E in non-decreasing order of edge weights.

$E_r \leftarrow \emptyset$. counter $\leftarrow 0$ // maintaining edge.

$k \leftarrow 0$ // pruned edge

why execute $<|V|-1$ do

$k \leftarrow k+1$

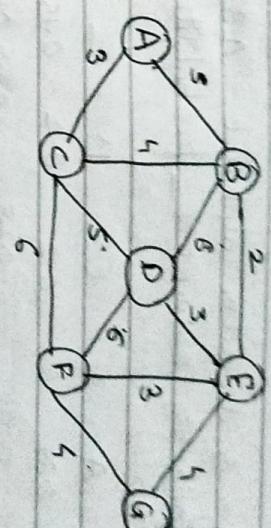
$E_r \leftarrow E_r \cup e_{ijk}$

counter \leftarrow counter + 1

return E_r .

HUFFMAN CODING

Data compression, Variable length coding
Fixed length coding.



$(B,E)^2, (A,C)^3, (D,F)^3, (A,B,E,F)^3, (E,G)^5$
 $(C,G)^4, (B,C)^4, (C,D)^5, (A,B)^5, (C,F)^6, (D,F)^6$

$$V = (1-2 \cdot 2)^2 \times 0.4 + (3-2 \cdot 2)^2 [0.1+0.2+0.15+0.15] = 0.96.$$

③ Construct a.

a	b	c	d	e	-
0.5	0.35	0.5	0.1	0.4	0.2

d	-	b	c	a	c
0.1	0.2	0.35	0.4	0.5	0.5

④ Construct a Huffman code for the following table.

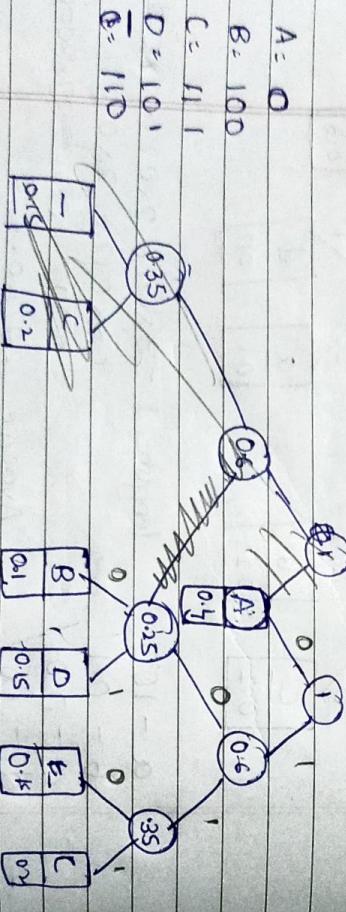
a	A	B	C	D	-
0.4	0.1	0.2	0.15	0.15	

0.25×0.35

\boxed{d}

\rightarrow B D - C | | A

0.1 0.15 0.15 0.2 0.1



c

Decide the best whose Encoding

1000101110001010

B A D - A D A

E - 110

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

~~1000101110001010~~

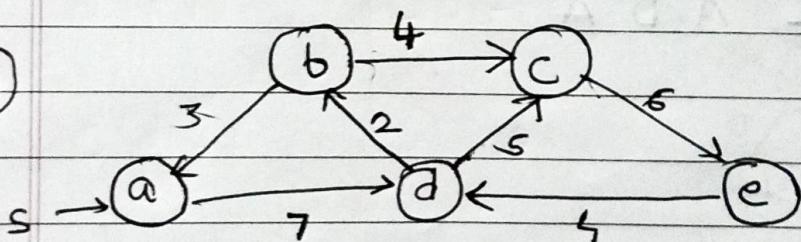
~~1000101110001010~~

<p

DIJKSTRA

Algorithm Dijkstra (G)for every vertex $v \in V$ do $d_v \rightarrow \infty$, $p_v \leftarrow \text{null}$.// $p_v = \text{Parent of } v$ Insert(Q, v, d_v) // Initialize vertex priority. $d_s \leftarrow 0$,Decrease(Q, s, d_s) // Update priority of $V_T \leftarrow \emptyset$ for $i \leftarrow 0$ to $|V| - 1$ do $u^* \leftarrow$ $V_T \leftarrow V_T \cup \{u^*\}$ for every vertex $u \in V - V_T$ adjacent to u^* if $d_{u^*} + w(u^*, u) < d_u$ $d_u \leftarrow d_{u^*} + w(u^*, u)$ $p_u \leftarrow u^*$ Decrease(Q, u, d_u)

1



ALGORITHM

$$h(g, 3) \quad i(h, 6) \quad k(h, 7)$$

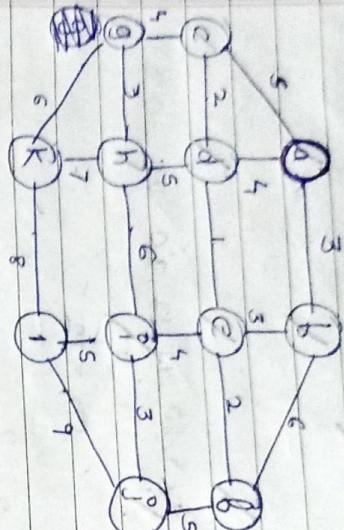
$$i^*(e, 4) \quad j^*(i^*, 3) \quad k^*(i^*, 5)$$

$$j^*(i^*, 3) \quad f^*(i^*, 5) \quad l^*(j^*, 9)$$

$$f^*(e, 2)$$

$$l(i^*, 5)$$

$$k(g, 6)$$

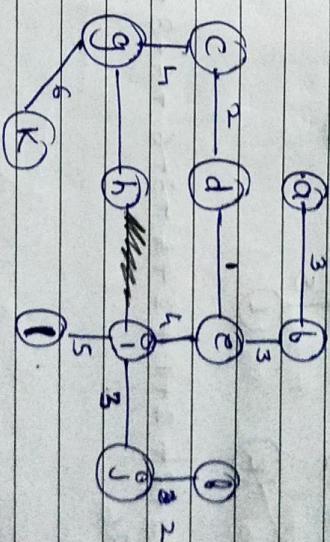


$$a(-, -) \quad b(a, 3) \quad c(g, 5), d(a, 4)$$

$$b(a, 3) \quad c(3b, 3), d(b, 6)$$

$$e(b, 3) \quad d(e, 1), i^*(e, 4) \quad f(e, 2)$$

$$d(c, 1) \quad c(d, 2), h(d, 5)$$



$$c(d, 2) \quad g(c, 4)$$

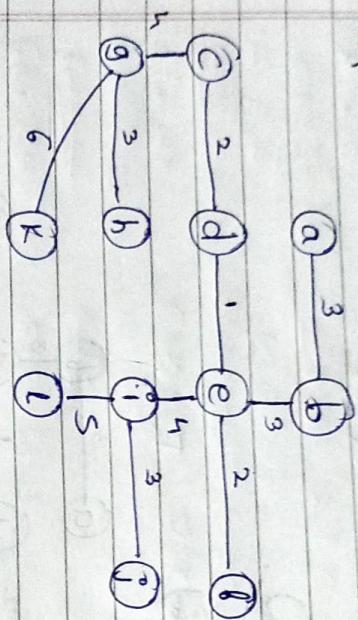
$$g(c, 5) \quad h(g, 3), k(g, 6)$$

$$= 3 + 3 + 1 + 4 + 2 + 3 + 4 + 3 + 2 + 5 + 6$$

$$= 36$$

Kruskal

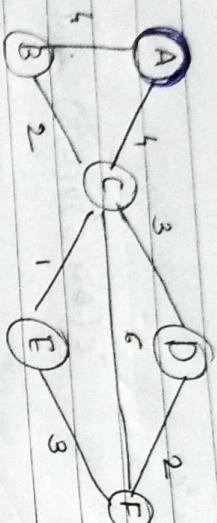
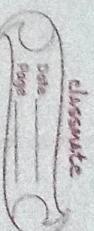
$$(d,e)^1, (c,d)^2, (e,f)^2, (g,h)^3, (i,j)^3
(b,c)^3, (a,b)^3, (a,d)^4, (c,i)^4, (c,g)^4
(d,k)^5, (a,c)^5, (b,j)^5, (i,l)^5
(b,l)^6, (h,i)^6, (g,k)^6, (h,k)^7, (k,l)^8, (l,j)^9$$



$$= 3 + 3 + 2 + 1 + 2 + 5 + 3 + 6 + 4 + 3 + 5 \\ \geq 36$$

Dijkstra's

$\alpha(-, 0)$ $c(a, a, s)b(a, 3), d(a, 4)$
 $b(a, 3)$ $e(b, 6), f(b, 9)$
 $e(b, 6)$ $b($


 $\Delta(-, 0)$
 $C(A, 4) \quad B(A, 5)$
 $v^* = C, \quad v \rightarrow D, E, F$

$A \rightarrow B = 5 \quad d[B] = 0 + 5 = 5$

$A \rightarrow C = 5 \quad d[C] = 0 + 5 = 5$

$A \rightarrow D \Rightarrow A \rightarrow C \rightarrow D = 7 \quad d[D] = 5 + 2 = 7$

$A \rightarrow E \Rightarrow A \rightarrow C \rightarrow E = 5$

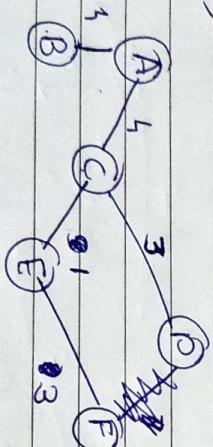
$A \rightarrow F = A \rightarrow C \rightarrow E \rightarrow F = 8 \quad v^* = E \quad v \rightarrow F$

$d[F] = 5 + 3 = 8$

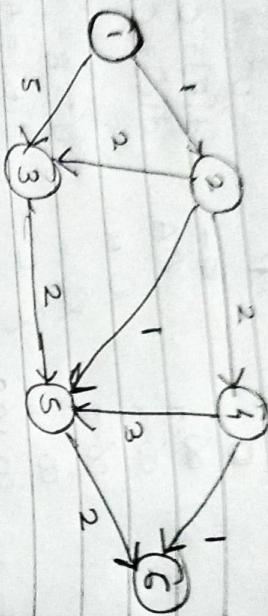
 $C(A, 5) \quad O(C, 7) \quad E(C, 5) \quad F(C, 8)$
 $D(C, 7)$
 $F(E, 8) \quad D(F, 10)$

$v^* = F \quad v \rightarrow D$

$d[D] = 8 + 2 = 10$

 $D(C, 7)$


$= 4+5+8+7 = 28$



$$\begin{aligned}
 1 &\rightarrow 2 = 1 \\
 1 &\rightarrow 3 = 1 \rightarrow 2 \rightarrow 3 = 3 \\
 1 &\rightarrow 4 = 1 \rightarrow 2 \rightarrow 4 = 3 \\
 1 \rightarrow 5 &= 1 \rightarrow 2 \rightarrow 5 = 2 \\
 1 \rightarrow 6 &= 1 \rightarrow 2 \rightarrow 5 \rightarrow 6 = 4
 \end{aligned}$$

③ 1(1, 0) 2(1, 1) 3(1, 5)

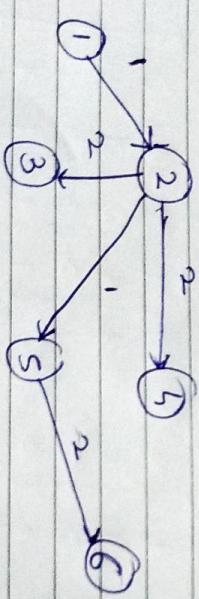
• 2(1, 1) 4(2, 3), 5(2, 2) 3(2, 3)

5(2, 2) 6(5, 2)

6(5, 5)

4(2, 3)

3(2, 3)



Items	A	B	C	D
Profit	280	100	120	120
Weight	40	10	20	24
Capacity = 60				

→ $\frac{\text{Profit}}{\text{Weight}}$ → 70 → 10 → 6 → 5

$$C = 60 - 10 = 50 - 40 = 10$$

$$W = 10 + 40 + 10 = 60$$

$$P = 100 + 280 + \frac{10}{20} \times 120 = 540$$

$$\left[\begin{array}{cccc}
 x_1 & x_2 & x_3 & x_4 \\
 1 & 1 & 0.5 & 0 \\
 10/20 & & &
 \end{array} \right]$$

• Objects	1	2	3	4	5	6	7
Capacity	5	10	15	7	8	9	4
Profit	1	3	5	4	1	3	2
Weight	= 15						