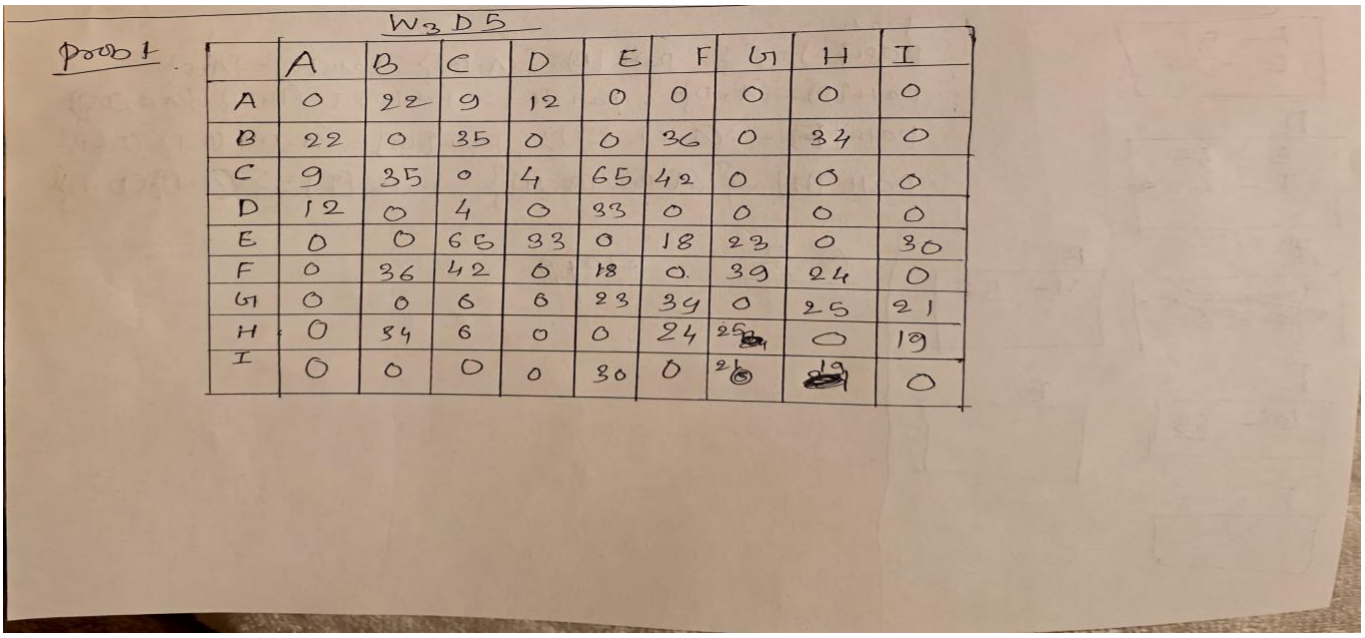
W3D5_solution

Q1

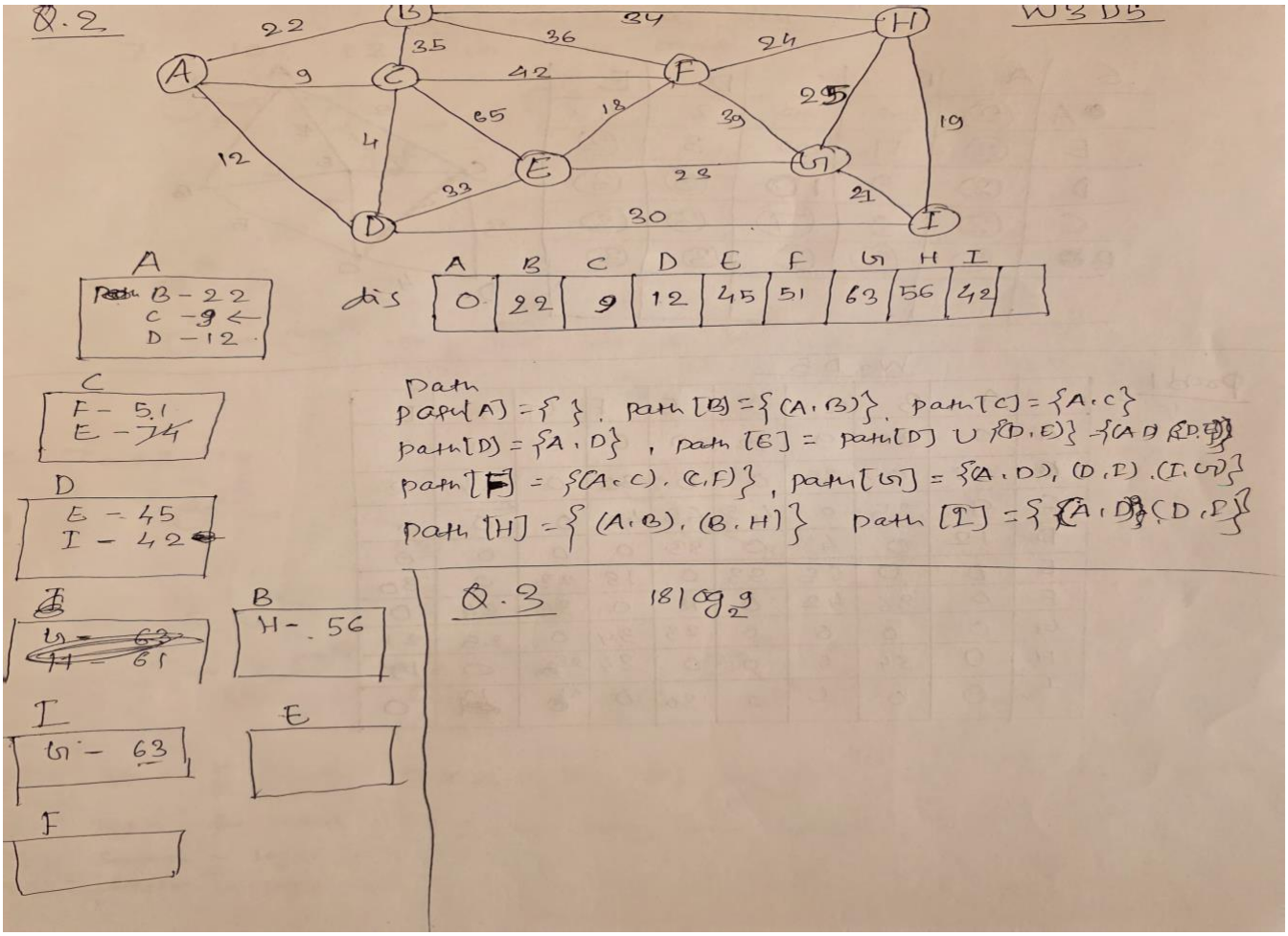


W3 D5

Proof

	A	B	C	D	E	F	G	H	I
A	0	22	9	12	0	0	0	0	0
B	22	0	35	0	0	36	0	34	0
C	9	35	0	4	65	42	0	0	0
D	12	0	4	0	33	0	0	0	0
E	0	0	65	33	0	18	23	0	30
F	0	36	42	0	18	0	39	24	0
G	0	0	6	0	23	39	0	25	21
H	0	34	6	0	0	24	25	0	19
I	0	0	0	0	30	0	21	19	0

Q2,3



Q.2

W3 D5

A
B - 22
C - 9 ←
D - 12

C
F - 51
E - 74

D
E - 45
I - 42

G
H - 56

I
G - 63

B
H - 56

E

F

	A	B	C	D	E	F	G	H	I
dis	0	22	9	12	45	51	63	56	42

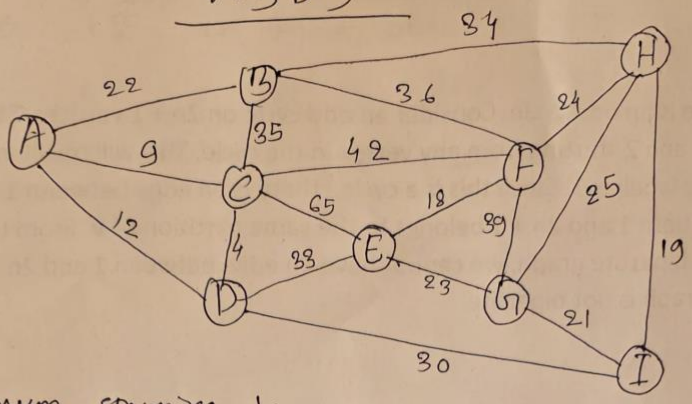
Path
path[A] = { }, path[B] = {(A, B)}, path[C] = {A, C}
path[D] = {A, D}, path[E] = path[D] ∪ {(D, E)} = {(A D) (D E)}
path[F] = {(A, C), (C, F)}, path[G] = {(A, D), (D, I), (I, G)}
path[H] = {(A, B), (B, H)} path[I] = {(A, D), (D, I)}

Q.3 $18 \log_2 9$

Q4,5

W3D5

Q. 4.



Step 1 Minimum spanning tree using Kruskal's Algorithm.

Edges

C, D – 4
A, C – 9
A, D – 12
E, F – 18
H, I – 19
G, I – 21
A, B – 22
E, G – 23
F, H – 24
G, H – 25
D, I – 30
D, E – 33
B, H – 34
B, C – 35
B, F – 36
F, G – 39
C, F – 42
C, E – 65

Step 2

(A) (B) (C) (D) (E) (F) (G) (H) (I)

pick C,D — (A) (B) (CD) (E) (F) (G) (H) (I)

pick A,C — (ACD) (B) (E) (F) (G) (H) (I)

pick ~~A,D~~

pick E,F — (ACBD) (B) (EF) (G) (H) (I)

pick H,I — (ACD) (B) (EF) (G) (HI)

pick G,I — (ACD) (B) (EF) (GHI)

pick A,B — (ABCD) (EF) (GHI)

pick E,G — (ABCD) (EFGHI)

pick D,I — (ABCDEFGHI)

Minimum cost of spanning tree = 4 + 9 + 18 + 19 + 21 + 22 + 23 + 30

= 13 + 18 + 19 + 21 + 22 + 23 + 30

= 146 Ans.

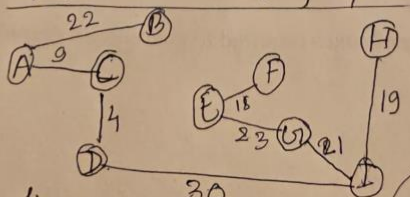


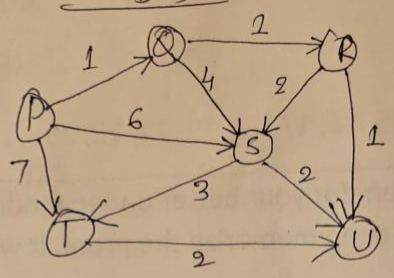
fig. minimum spanning tree.

Q.5 $O(18 \log_2 9)$

Q6,7

W3 Q5

Q6



Adjacency matrix of above graph

	P	Q	R	S	T	U
P	0	1	0	6	7	0
Q	0	0	1	4	0	0
R	0	0	0	2	0	1
S	0	0	0	0	3	2
T	0	0	0	0	0	2
U	0	0	0	0	0	0

Q7

Topological sort

P Q R S T U

dist

0	1	2	4	7	3	
P	Q	R	S	T	U	

dis[P] = 0 path = { }

dis[Q] = dis[P] + wt(P,Q) = 0+1 = 1 , path = {(P,Q)}

dis[R] = dis[Q] + wt(Q,R) = 1+1 = 2 , path = {(P,Q),(Q,R)}

dis[S] = dis[R] + wt(R,S) = 2+2 = 4 , path = {(P,Q),(Q,R),(R,S)}

dis[T] = dis[P] + wt(P,T) = 0+7 = 7 , path = {(P,T)}

dis[U] = dis[R] + wt(R,U) = 2+1 = 3 , path = {(P,Q),(Q,R),(R,U)}

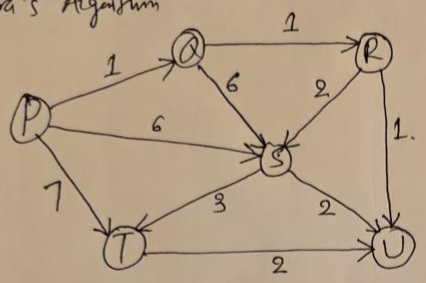
Q8,9,10

W3D5

Q.8 $O(10+6) = O(16)$

Q.9 Since the weights are all positive so we can apply Dijkstra's Algorithm

10.



P path[P] = { }

Distance	Q	R	S	T	U
	1	2	4	7	3

Q – 1 ←
T – 7
S – 64

path[Q] = path[P] ∪ {P, Q} = {(P, Q)}

Q
R – 2 ←

path[R] = path[Q] ∪ {Q, R} = {(P, Q)(Q, R)}

R
S – 4 ←
U – 3 ←

path[U] = path[R] ∪ {R, U} = {P, Q, (Q, R) (R, U)}

U

path[S] = path[R] ∪ {(R, S)} = {(P, Q), (Q, R), (R, S)}

S

path[T] = path[P] ∪ {(P, T)} = {(P, T)}

T