DAI Mengjia

One of the possible solutions.
Credit to Dai Menjia.

Assignment 3

CSCI 2100 B

6.2 (3) There are 9-1 = 8 passes.

After P=1 1 3 4 1 5 9 2 6 5

P=2 1 1 4 3 5 9 2 6 5

P=3 1 1 2 3 5 9 4 6 5

P=4 1 1 2 3 5 9 4 6 5

P=5 1 1 2 3 4 9 5 6 5

P=6 1 1 2 3 4 5 9 6 5

P=7 1 1 2 3 4 5 5 6 9

P=8 1 1 2 3 4 5 5 6 9

6.3

7-Sort: 2 1 7 6 5 4 3 9 8

3-Sort: 2 1 4 3 5 7 6 9 8

1-Sort: 1 2 3 4 5 6 7 8 9

6.4

Because for mergesort, it has to do the same many number of comparisons no matter the input sequence, the input sequence does not matter too much.

Let $T(n)$ denote the time needed to mergesort n inputs

Then, $\begin{cases} T(n) = 2T(\frac{n}{2}) + n \\ T(1) = 1 \end{cases}$

$\Rightarrow T(n) = n\log(n) + n$

$\therefore$ The running time for sorted input, reverse-ordered input and random input is $n\log(n) + n$ when using mergesort.

6.6

(2) ~~After~~ P=1, pivot = median(3, 5, 9) = 5

$\therefore$ After P=1, it becomes

3 1 4 1 3 5 3 ⋮ 5 9 5 6.

P=2, for the left part,

pivot = median(3, 3, 1) = 3

$\Rightarrow$ 3 1 1 ⋮ 3 3 5 4

for the right part,

pivot = median(5, 6, 9) = 6

$\Rightarrow$ 5 5 ⋮ 6 9

$\Rightarrow$ 3 1 1 ⋮ 3 3 5 4 ⋮ 5 5 ⋮ 6 9

P=3, for the left part

$\Rightarrow$ 1 1 3

for the middle part

$\Rightarrow$ 3 3 4 5

for the right part

$\Rightarrow$ 5 5 6 9

$\Rightarrow$ Sorted sequence:

1 1 3 3 3 4 5 5 6 9

6.7 The sorted and reverse-ordered input have the same time complexity because the median-of-three partioning will give the same pivot that is the median of the sequence.

Assume the two subarrays are exactly half the size of the original. And we discuss the Big(O) answer.

Let $T(N)$ denote the time.

n, $T(N) = 2T(\frac{N}{2}) + cN$

c is a constant

$\Rightarrow \frac{T(N)}{N} = \frac{T(\frac{N}{2})}{\frac{N}{2}} + c$

$\Rightarrow T(N) = cN\log N + N = O(N\log N)$.

For random input.

Let S_1, S_2 be two subarrays.

Then size for S_1 is equally likely with a probability $\frac{1}{N}$.

$T(N) = T(i) + T(N-i-1) + cN$

$\because E(T(i)) = E(T(N-i-1)) = \frac{1}{N}\sum_{j=0}^{N-1} T(j)$

$\therefore T(N) = \frac{2}{N}\left[\sum_{j=0}^{N-1} T(j)\right] + cN$

$\Rightarrow NT(N) = 2\left(\sum_{j=0}^{N-1} T(j)\right) + cN^2$

$\Rightarrow (N-1)T(N-1) = 2\left(\sum_{j=0}^{N-2} T(j)\right) + c(N-1)^2$

Do the subtraction

$\Rightarrow NT(N) - (N-1)T(N-1) = 2T(N-1) + 2cN - c$

$\Rightarrow \frac{T(N)}{N+1} = \frac{T(N-1)}{N} + \frac{2c}{N+1}$

$\Rightarrow T(N) = O(N\log N)$.

6.9

Stable: Bubble sort and Merge sort.

Unstable: Selection sort and Quick sort.

7-1

Indegree before Dequeue #

Vertex	1	2	3	4	5	6	7	8	9	10	11
S	0	0	0	0	0	0	0	0	0	0	0
A	2	1	1	0	0	0	0	0	0	0	0
D	2	1	0	0	0	0	0	0	0	0	0
G	1	0	0	0	0	0	0	0	0	0	0
B	1	1	1	1	1	0	0	0	0	0	0
E	4	4	3	2	1	0	0	0	0	0	0
H	1	1	0	0	0	0	0	0	0	0	0
C	3	3	3	3	3	3	2	1	1	0	0
F	2	2	2	2	2	2	2	1	0	0	0
I	2	2	2	2	1	1	1	0	0	0	0
t	3	3	3	3	3	3	3	3	2	1	0
Enqueue	S	G	D	H	A	B	E	I	F	C	t
Dequeue	S	G	D	H	A	B	E	I	F	C	t

So, the topological ordering is:

SGDHABEIFCt

7-3 (1)

After A is known

V	known	dv	pv
A	1	0	0
B	0	5	A
C	0	3	A
D	0	—	0
E	0	—	0
F	0	—	0
G	0	—	0

After C is known.

V	known	dV	pv
A	1	0	0
B	0	5	A
C	1	3	A
D	0	10	C
E	0	10	C
F	0	—	0
G	0	—	0

...er (B) is known. ⇒ After (G) is known

V	known	dv	pv
A	1	0	0
B	1	5	A
C	1	3	A
D	0	10	C
E	0	8	B
F	0	-	0
G	0	6	B

V	known	dv	pv
A	1	0	0
B	1	5	A
C	1	3	A
D	0	10	C
E	0	7	G
F	0	-	0
G	1	6	B

After (E) is known ⇒ After (F) is known

V	known	dv	pv
A	1	0	0
B	1	5	A
C	1	3	A
D	0	9	E
E	1	7	G
F	0	8	E
G	1	6	B

V	known	dv	pv
A	1	0	0
B	1	5	A
C	1	3	A
D	0	9	E
E	1	7	G
F	1	8	E
G	1	6	B

After (D) is known

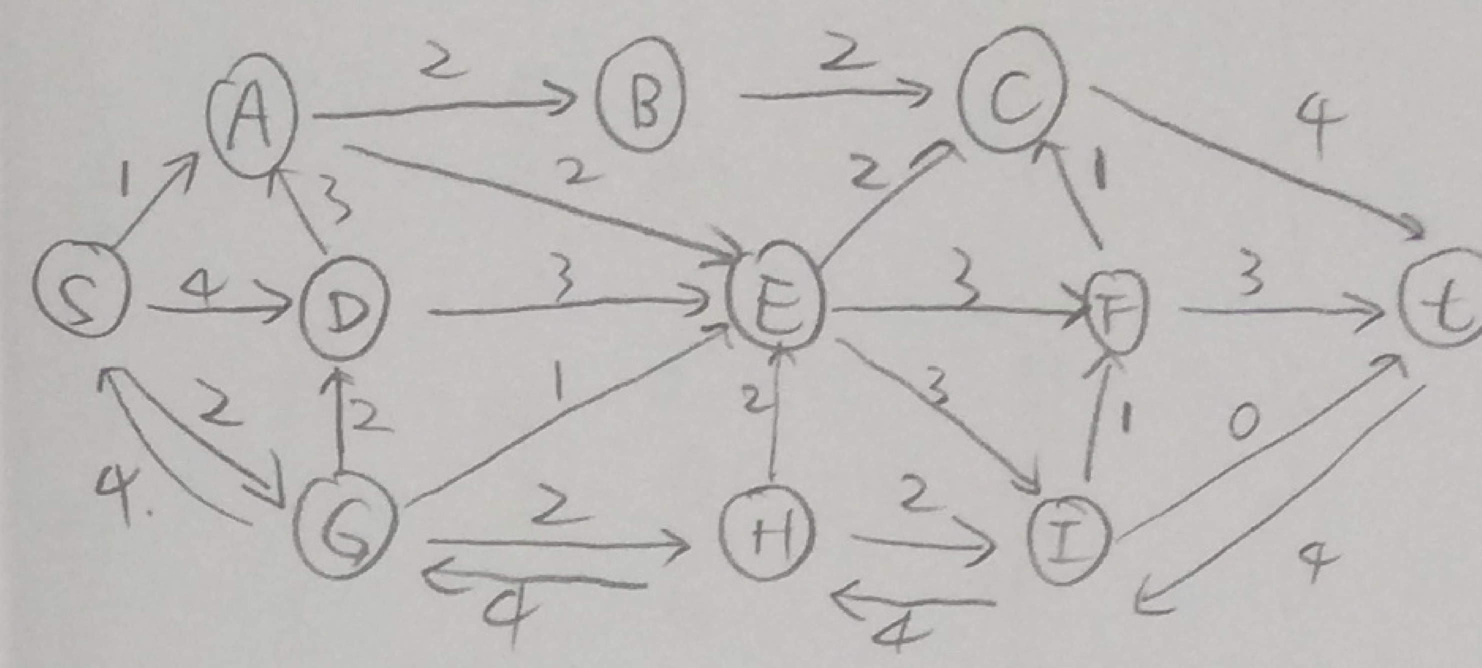
V	known	dv	pv
A	1	0	0
B	1	5	A
C	1	3	A
D	1	9	E
E	1	7	G
F	1	8	E
G	1	6	B

∴ The distance from A to other vertices:

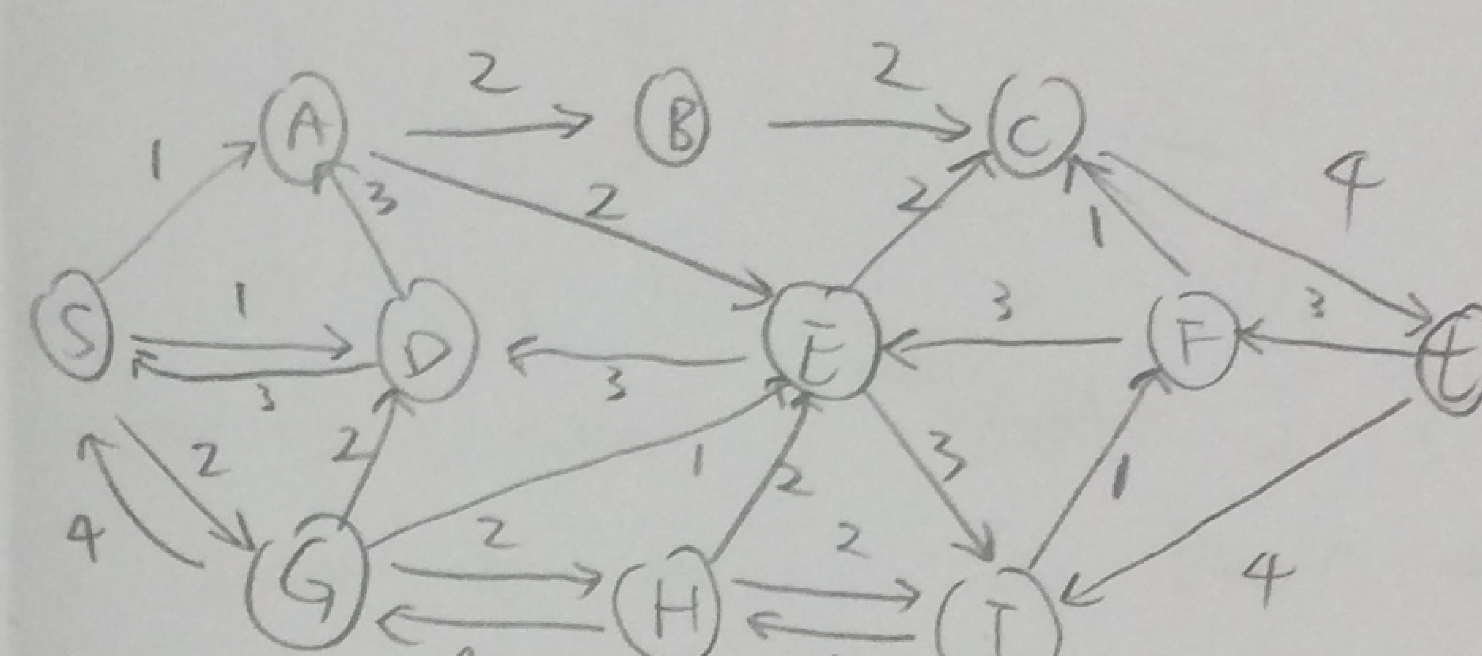
B	C	D	E	F	G
5	3	9	7	8	6.

7.4

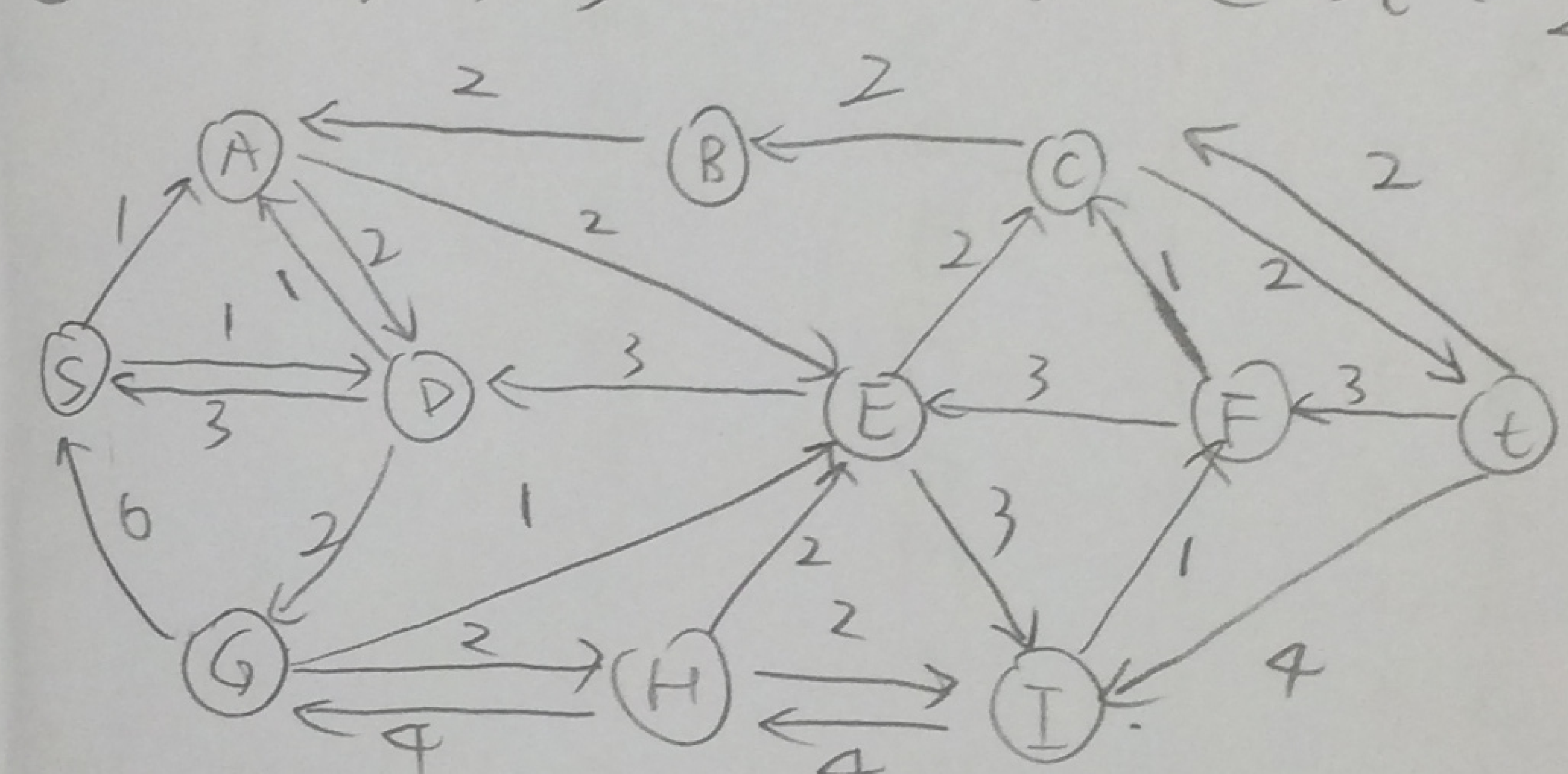
① $S \to G \to H \to I \to t$: 4



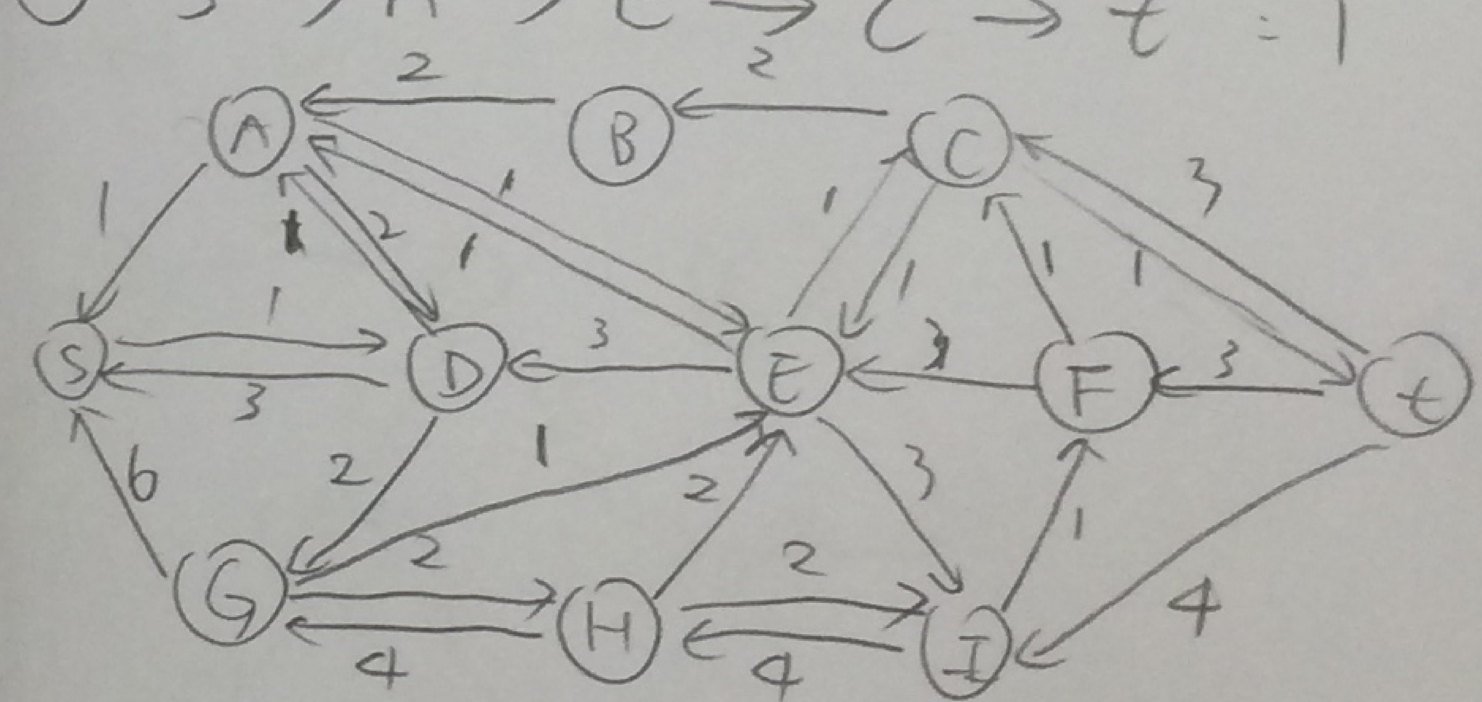
② $S \to D \to E \to F \to t$: 3



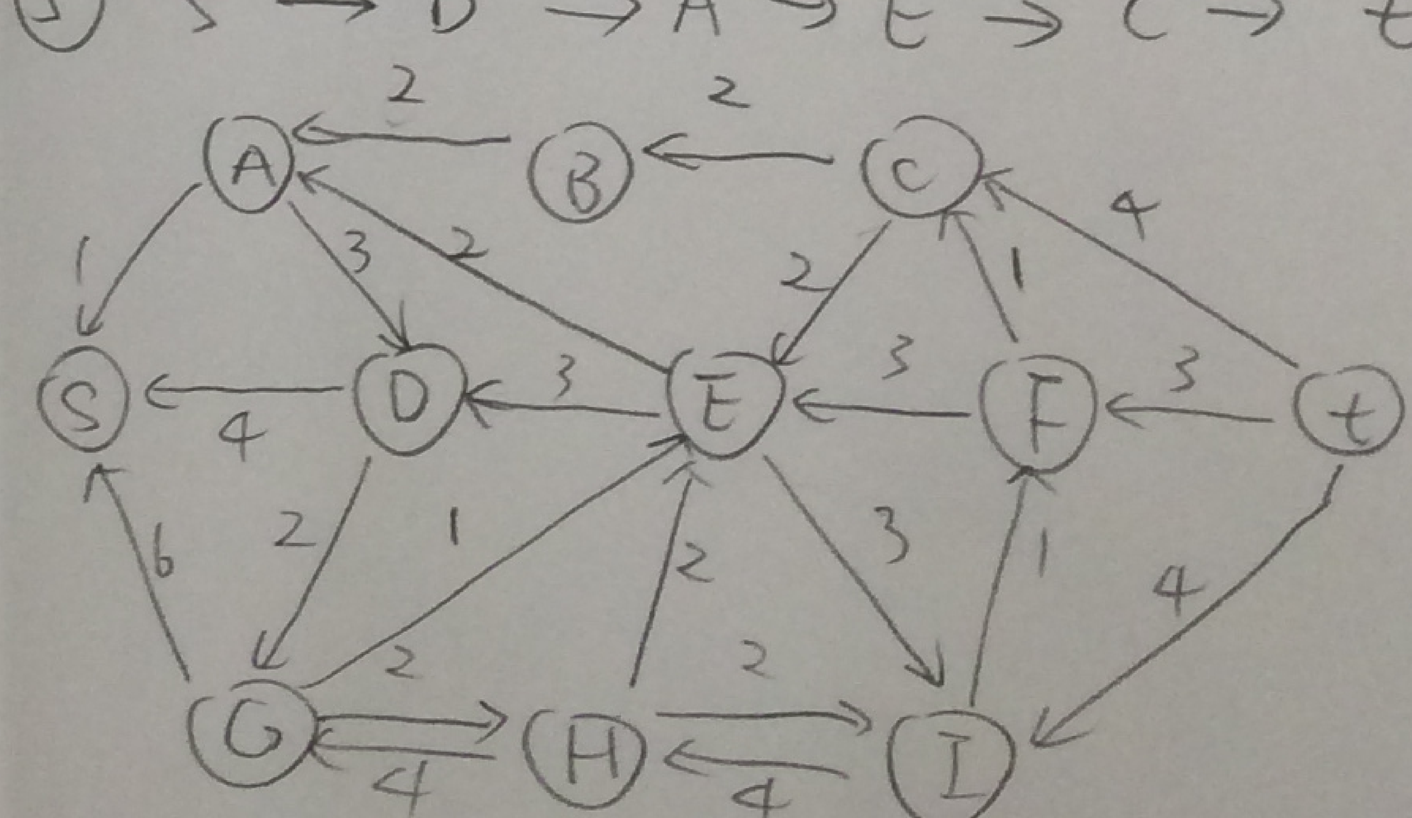
③ $S \to G \to D \to A \to B \to C \to t$: 2



④ $S \to A \to E \to C \to t$: 1



⑤ $S \to D \to A \to E \to C \to t$: 1



∴ maximum flow: 4+3+2+1+1 = 11

(1)

Prim's Algorithm

After A is known

V	known	dv	pv
A	1	0	0
B	0	3	A
C	0	—	0
D	0	4	A
E	0	4	A
F	0	—	0
G	0	—	0
H	0	—	0
I	0	—	0
J	0	—	0

⇒

After B is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	0	10	B
D	0	4	A
E	0	2	B
F	0	3	B
G	0	—	0
H	0	—	0
I	0	—	0
J	0	—	0

After E is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	0	10	B
D	0	4	A
E	1	2	B
F	0	3	B
G	0	—	0
H	0	2	E
I	0	1	E
J	0	—	0

⇒

After I is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	0	10	B
D	0	4	A
E	1	2	B
F	0	3	B
G	0	—	0
H	0	2	E
I	1	1	E
J	0	7	I

After H is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	0	10	B
D	0	4	A
E	1	2	B
F	0	3	B
G	0	—	0
H	1	2	E
I	1	1	E
J	0	7	I

⇒

After F is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	0	6	F
D	0	4	A
E	1	2	B
F	1	3	B
G	0	2	F
H	1	2	E
I	1	1	E
J	0	7	I

After G is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	0	1	G
D	0	4	A
E	1	2	B
F	1	3	B
G	1	2	F
H	1	2	E
I	1	1	E
J	0	7	I

⇒

After C is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	1	1	G
D	0	4	A
E	1	2	B
F	1	3	B
G	1	2	F
H	1	2	E
I	1	1	E
J	0	7	I

After D is known

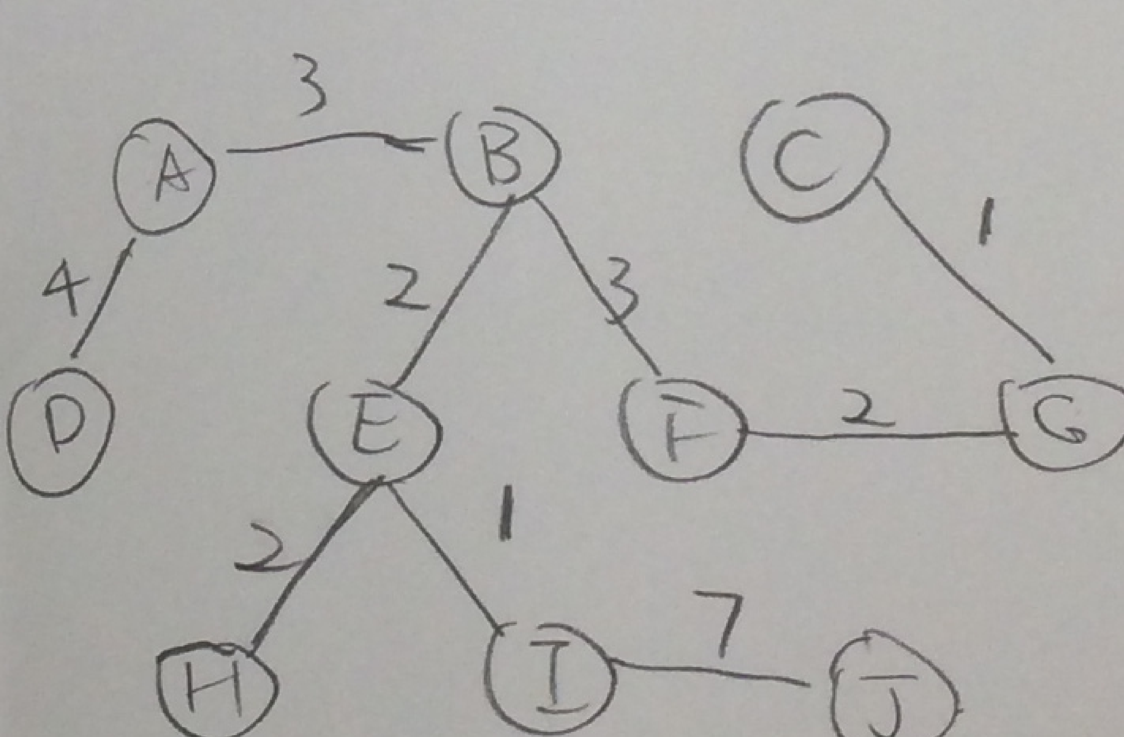
V	known	dv	pv
A	1	0	0
B	1	3	A
C	1	1	G
D	1	4	A
E	1	2	B
F	1	3	B
G	1	2	F
H	1	2	E
I	1	1	E
J	0	7	I

⇒

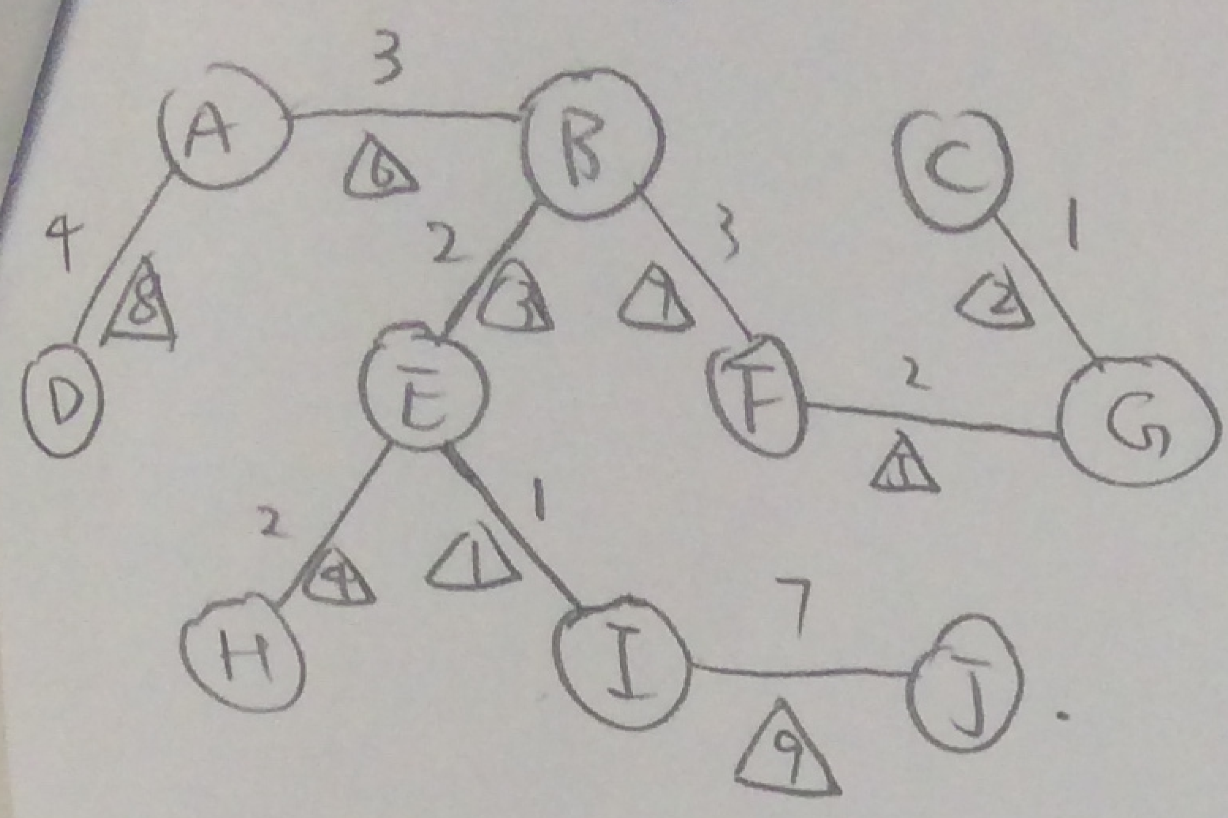
After J is known

V	known	dv	pv
A	1	0	0
B	1	3	A
C	1	1	G
D	1	4	A
E	1	2	B
F	1	3	B
G	1	2	F
H	1	2	E
I	1	1	E
J	1	7	I

Final configuration :



Kruskal's Algorithm



Notes: "①→⑨" means step 1 to step 9