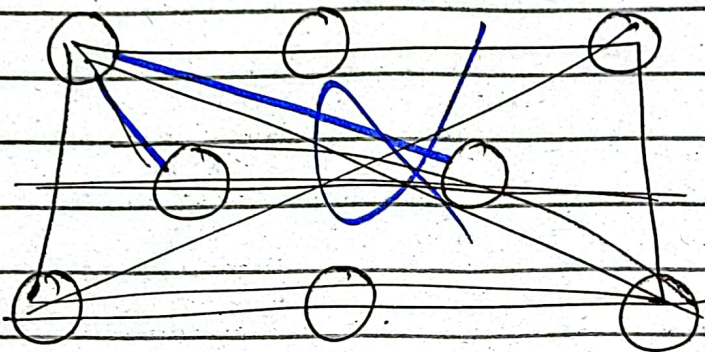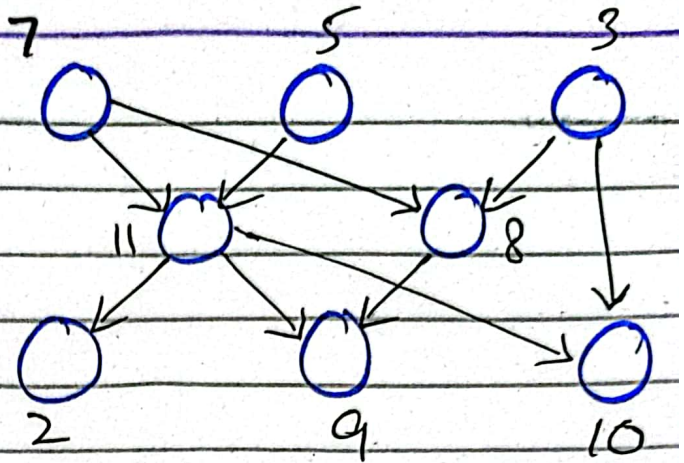Q No 1:

```
Search (Arr, size){
    int    left = 0;
    int    right = size +1;
    while(left ≤ right){

        mid = left + (right -left)/2;

        if (Arr[mid] = mid){
             return mid;
        }

        else if (Arr[mid] < mid){
             left = mid +1;
        }

        else {
             right = mid - 1;
        }

    }

    return -1;

}
```
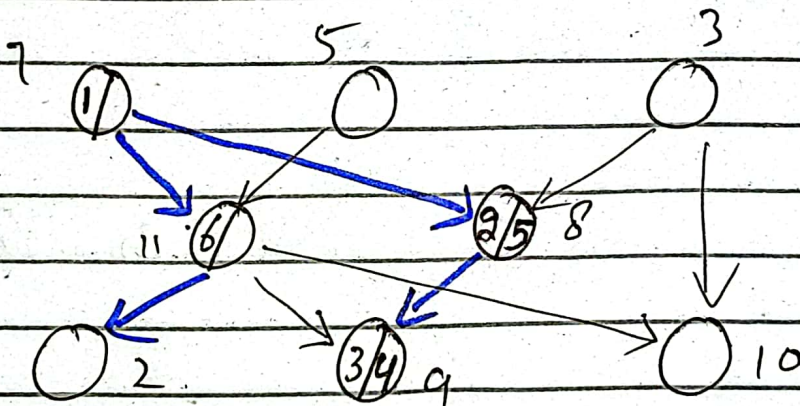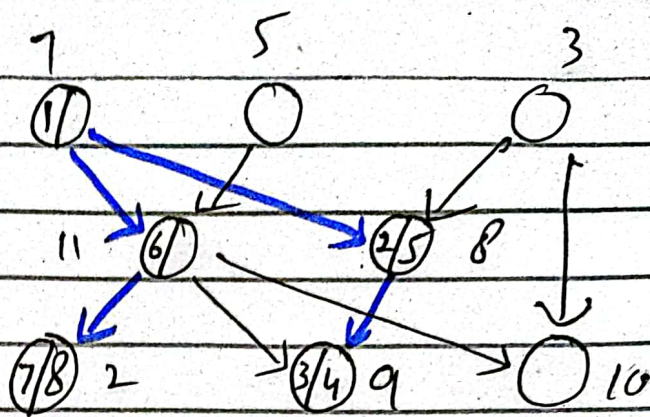
Q No. 2:

DFS.

Source Node is 7:

$\bigcirc$ = White

$\bigcirc\!\!\!\!/$ = Greg

$\bigcirc\!\!\!\!/\!\!/$ = Black.

(b) When DFS-VISIT Called for 5ᵗʰ time
(Just Called)

	Λ	11	Nil	Nil	Nil	7	8	Nil	7
Node	2	3	5	7	8	9	10	11	
d.time	-	-	-	1	2	3	-	6	
f.time	-	-	-		5	4	-	=	
Color	W	W	W	G	B	B	W	G	

Graph diagram with nodes labeled 7, 5, 3 at top; 11, 6/1, 2/5, 8 in middle; 7/8, 2, 3/4, 9, 10 at bottom.

(c) When L10 completed its work for 3rd time:

π	11	Nil	Nil	Nil	7	8	Nil	7
Node	2	3	5	7	8	9	10	11
d·time	7	–	–	1	2	3	–	6
f·time	8	–	–	–	5	4	–	–
Color	B	W	W	G	B	B	W	G

Graph diagram: nodes 7 (1/12), 5 (15/16), 3 (13/14), 11, 6/11, 2/5, 8, 7/8, 3/4, 9/10, with labels C2, C4, C3, C1.
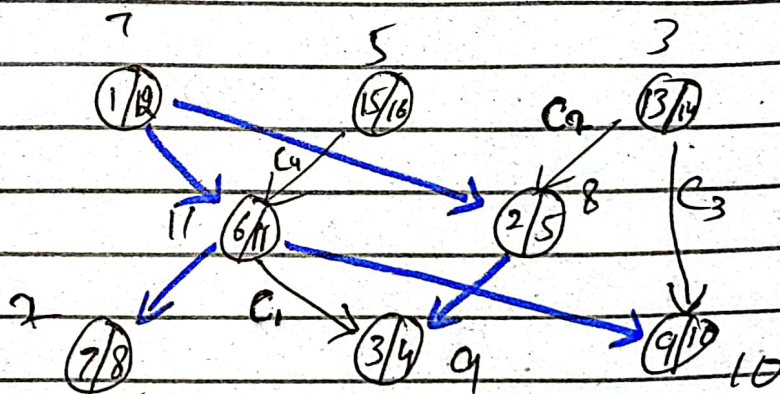
(a) Tree Edges

(7→8), (7,11)
(8→9), (11,10)
(11→8)

Cross Edges:

(11→9), (3→8)
(3→10), (5→11)

π	11	Nil	Nil	Nil	7	8	11	7
Node	2	3	5	7	8	9	10	11
d.time	7	13	15	1	2	3	9	6
f.time	8	14	16	12	5	4	10	11
Color	B	B	B	B	B	B	B	B

"――――――――"

Q No 3:
Dijkstra:
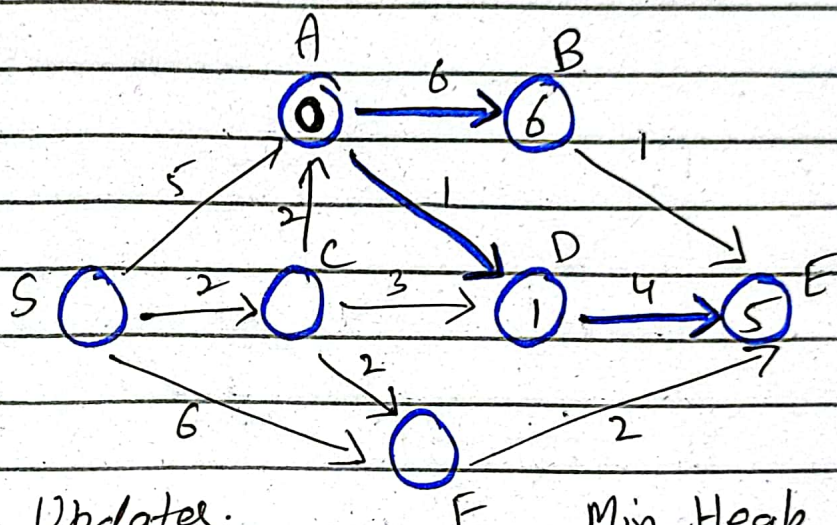
a) First 8th Updates:

Not possible: Only three
updates occured in the
given Graph.

Min Heap
A = 0
B = 6
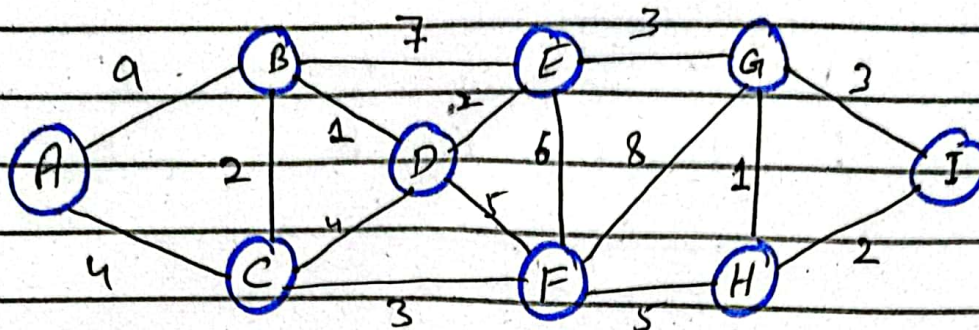C = ∞
D = 1
E = 5
F = ∞
S = ∞

b) Line 10 executed 8th time:
Not possibly: line 10 executed
only three times and after
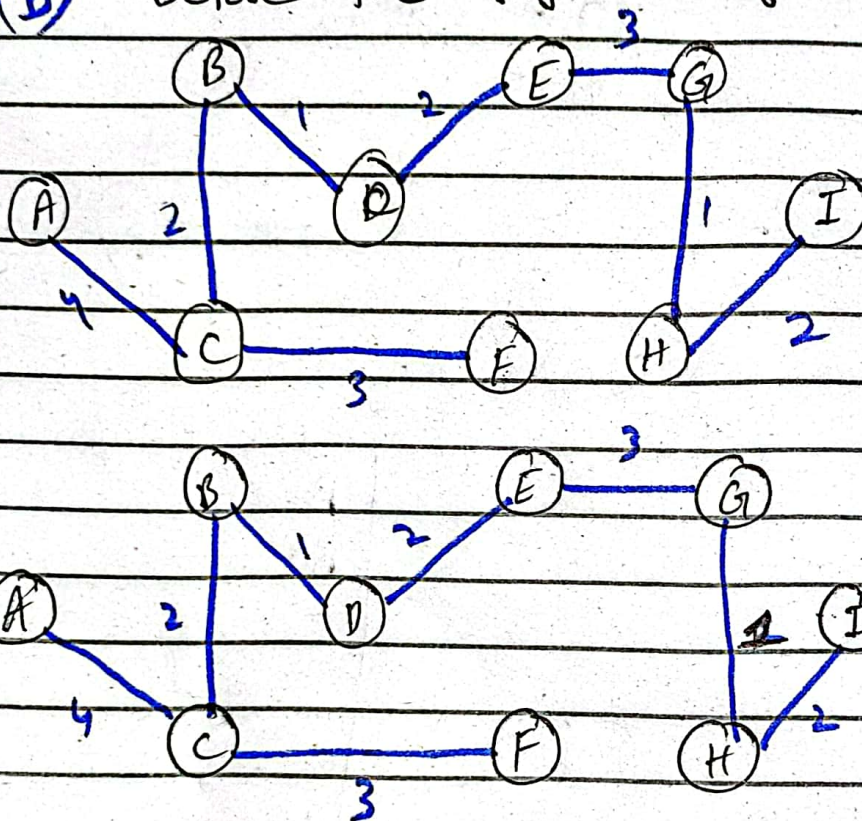executing 3 times and the nodes
which are not in Heap are:

π	Nil	A	D	A
Node	A	D	E	B
d	0	1	5	6

QNo 5:

MST-KRUSKAL:

(b) Before the rejection of 2nd edge.

BD→1	✓
GH→1	✓
BC→2	✓
DE→2	✓
HI→2	✓
CF→3	✓
EG→3	✓
GI→3	✗
AC→4	✓
CD→4	✗
DF→5	✗
FH→5	✗
EF→6	✗
BE→7	✗
FG→8	✗
AB→9	✗

c) Rejected Edge:
GI, CD, DE, FH,
EF, BE, FG, AB

(a) 1+1 + 2 + 2 +2 + 3 +3 + 4
Total weight is = 18

(d)   When   line   8   is   exicuted   for
6th   time: