Spring 2022

Q1:

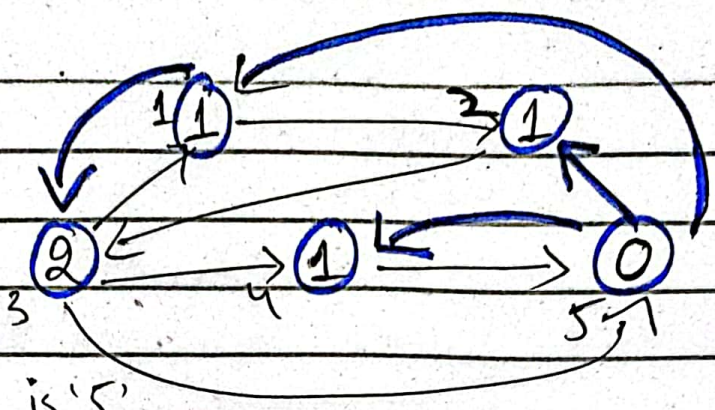
```
Sort (Arr, n){
    int countR, countG, countB = 0
n+1 ← for (int i=0; i<n; i++){
        if (Arr[i] == 'R'){
            countR++;
        }
        else if (Arr[i] == 'G'){
            countG++;
        }
        else if (Arr[i] == 'B'){
            countB++;
        }
    }
countR ← for (int i=0; i<countR; i++){
        Arr[i] = 'R';
    }
countG ← for (int j = countR; j<countG; j++){
        Arr[j] = 'G';
    }
countB ← for (int k=countG; k<countB; k++){
        Arr[k] = 'B';
    }
}
```

n ←

O(n)

Q No 2:

BFS:



Source Node is '5'.

π	Nil	5	5	5	1
Node	~~5~~	~~1~~	~~2~~	~~4~~	~~3~~
d	0	1	1	1	2

b) When 117 executed for the 4th Time: Vertices in the Q are:

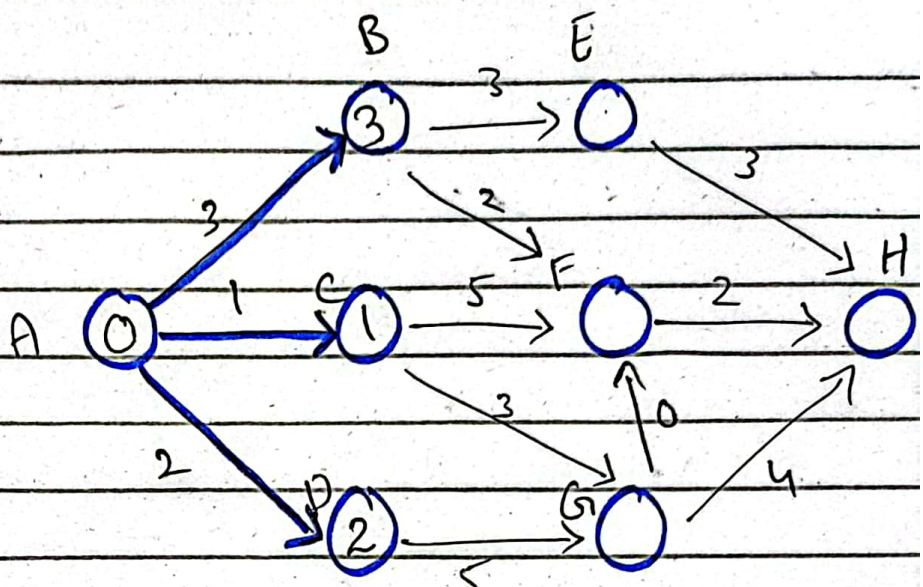
π	5	5	1
Node	2	4	3
d	1	1	2
C	Gr	Gr	Gr

a)

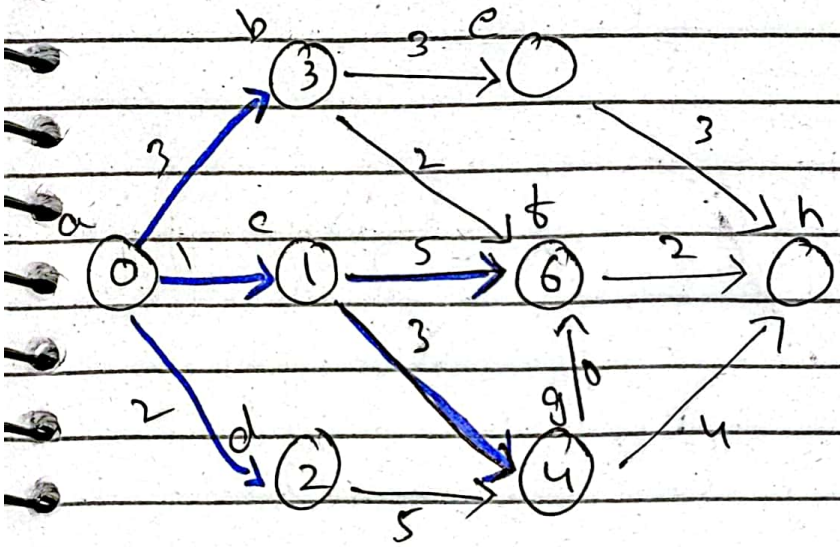
Tree Edges	Cross Edges
0→1	1→2
0→2	2→3
0→4	4→5
1→2	3→1
	3→4
	3→5

Q No 3:

Dijkstra:



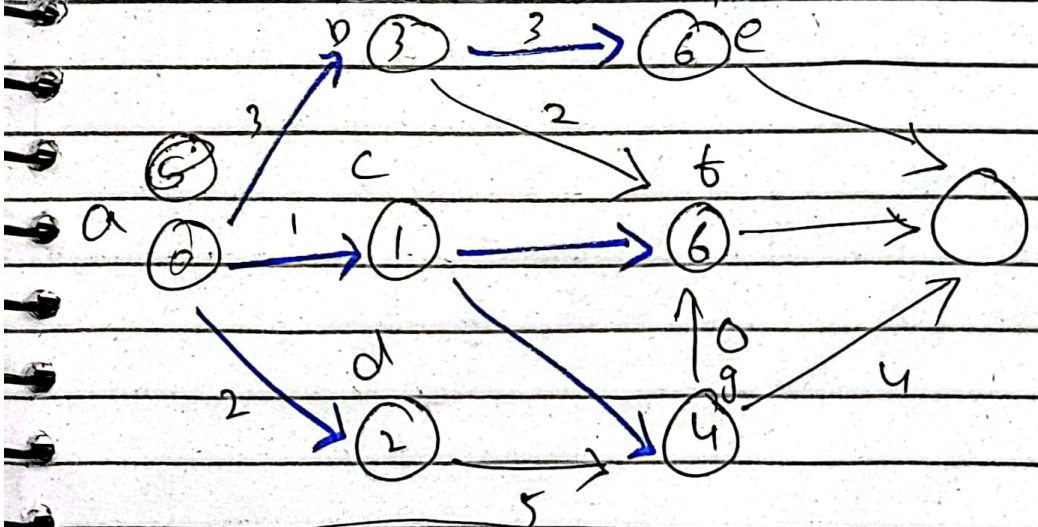
Line 10 executed 3 time.



Min Heap

~~a=0~~
~~b=3~~
~~c=1~~
~~d=2~~
~~e=6~~
~~f=6 5 4~~
~~g=4~~
~~h=8 6~~

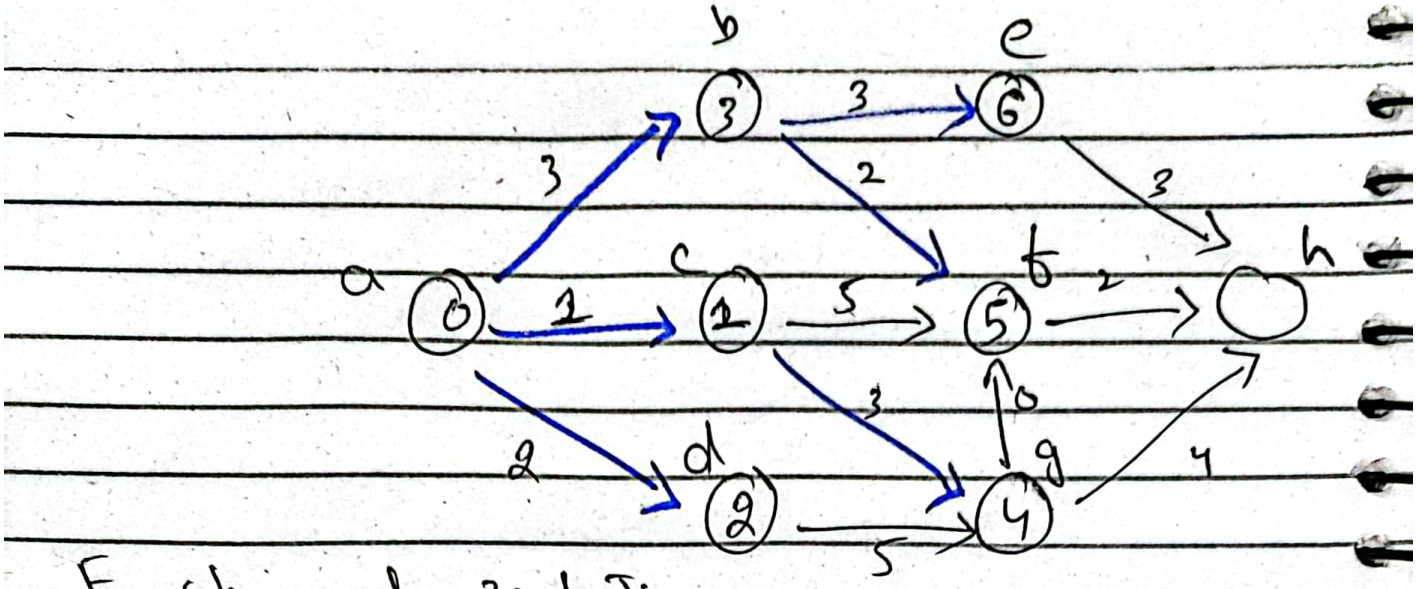
L10 executed 5th time.



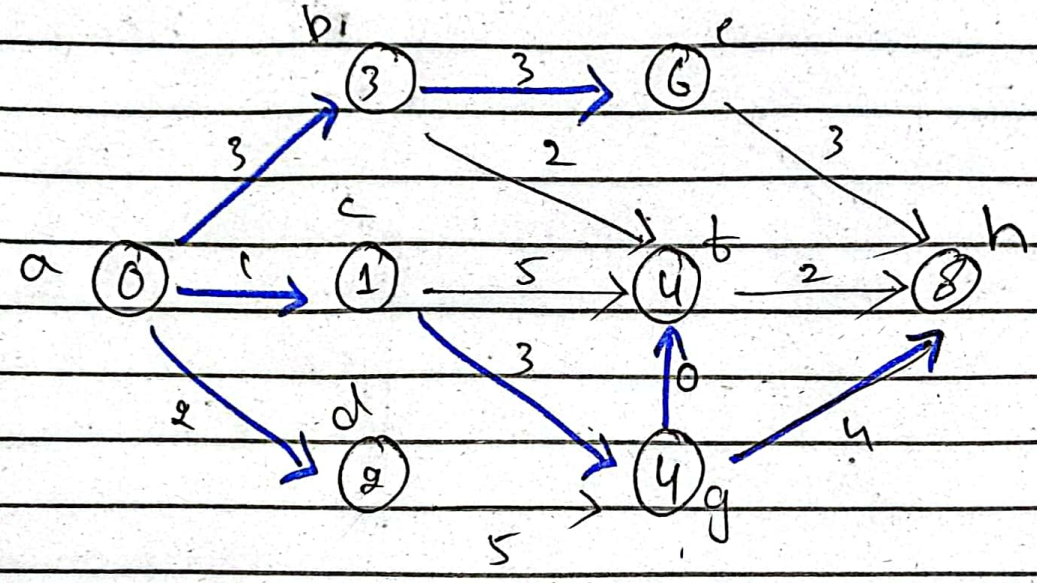
~~U = b~~ ~~V = e~~ (b)

~~U.d = 3~~ ~~V.d = 6~~

~~U.π = 0~~ ~~V.π = b~~



F changed 2nd Time.



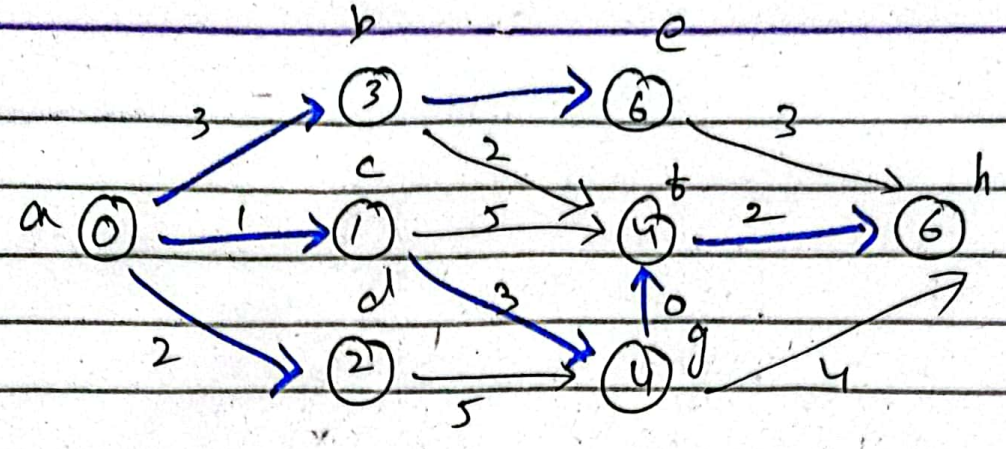
F changed for 3rd time.

(a)

$U = g$ $\quad V = f$

$U.\Pi = c$ $\quad V.\Pi = g$

$U.d = 4$ $\quad V.d = 4$



(b)

π	Nil	A	A
Node	A	C	B
d	0	1	3

π	Nil	a	a	a	b	g	c	f
Node	a	b	c	d	e	f	g	h
	0	∞	∞	∞	∞	∞	∞	∞
(a)	0	3	1	2	∞	∞	∞	∞
(c)	0	3	1	2	∞	6	4	∞
(d)	0	3	1	2	∞	6	4	∞
(b)	0	3	1	2	6	5	4	∞
(g)	0	3	1	2	6	4	4	8
(f)	0	3	1	2	6	4	4	6
(e)	0	3	1	2	6	4	4	6