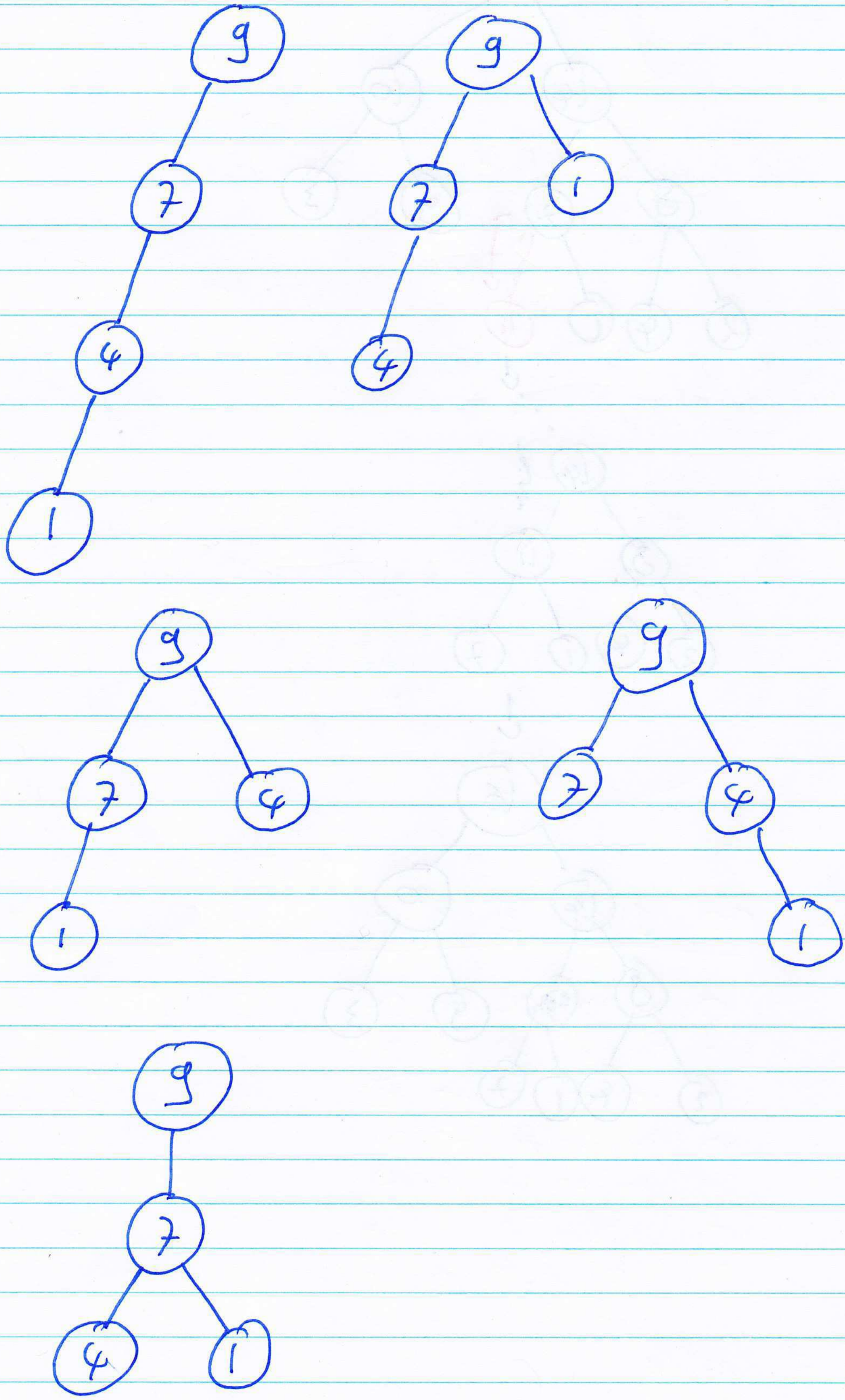
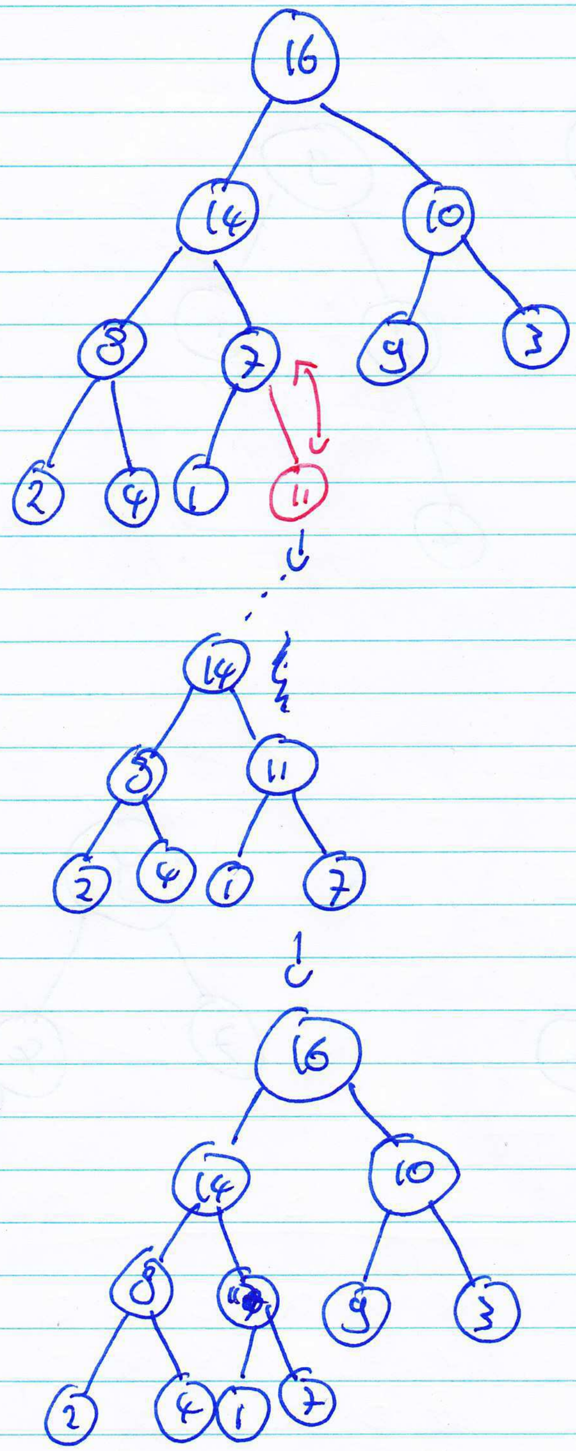
IV

1.

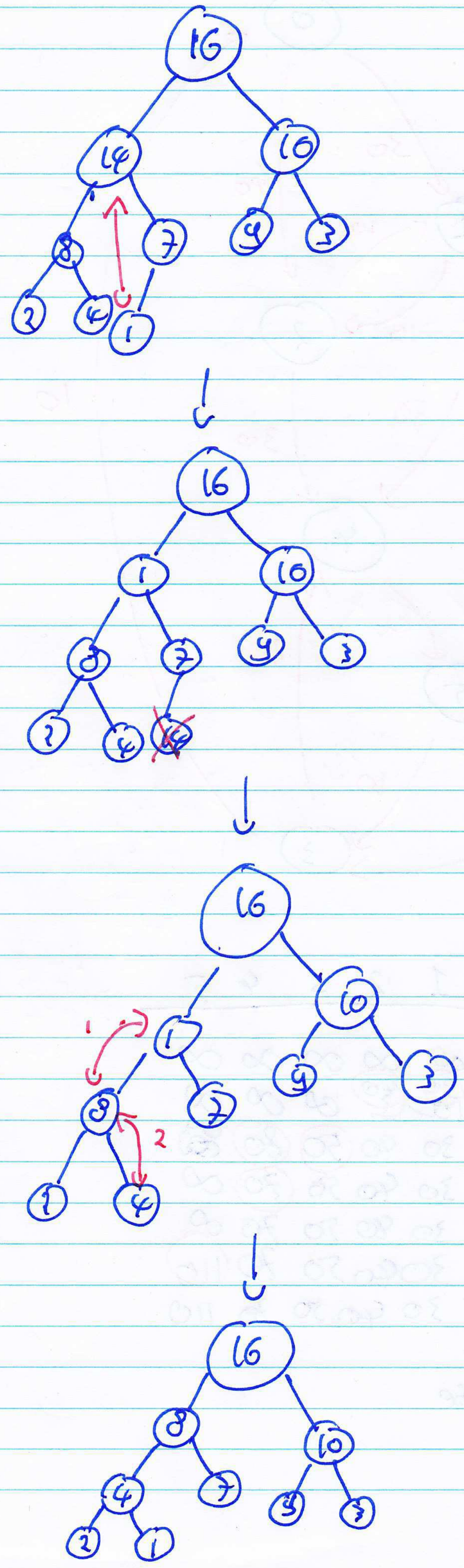


2

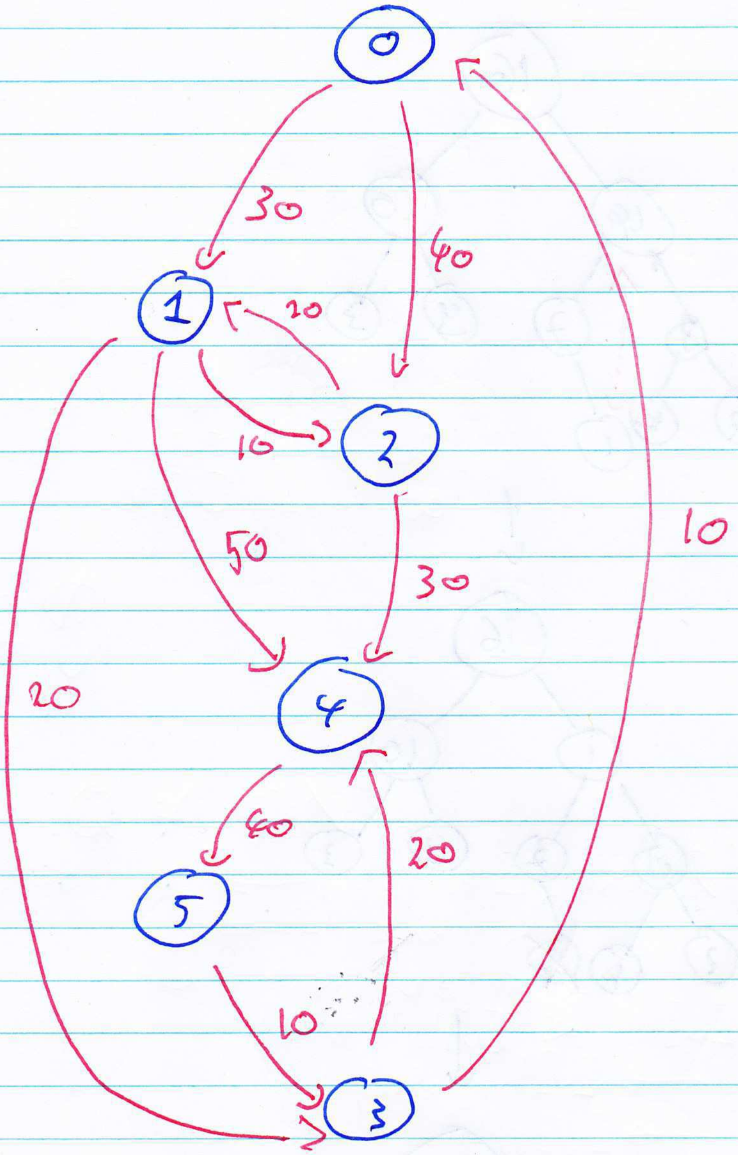
i



ii



3.



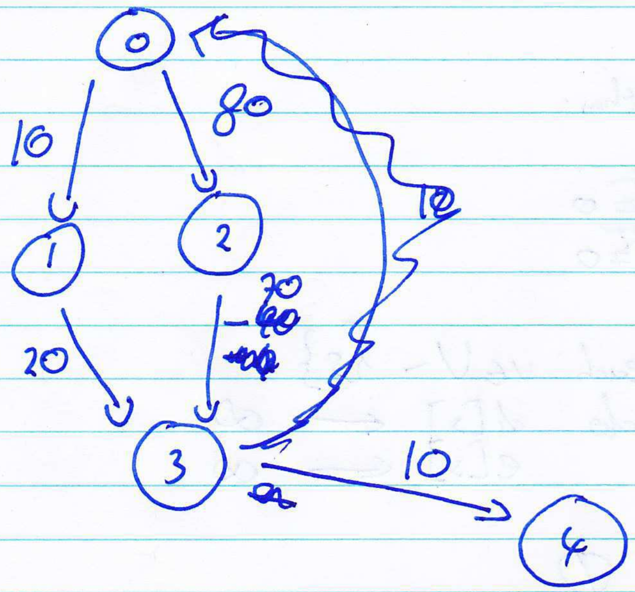
Q:	0	1	2	3	4	5
	0	∞	∞	∞	∞	∞
	0	30	40	∞	∞	∞
	0	30	40	50	80	∞
	0	30	40	50	70	∞
	0	30	40	50	70	∞
	0	30	40	50	70	110
	0	30	40	50	70	110

S : {0}
S : {0, 1}
S : {0, 1, 2}
S : {0, 1, 2, 3}
S : {0, 1, 2, 3, 4}
S : {0, 1, 2, 3, 4, 5}

x = update

4.

Graph



Q:

0	1	2	3	4	
0					
0	∞	∞	∞	∞	S: {0}
0	10	80	∞	∞	S: {0, 1}
0	10	80	30	∞	S: {0, 1, 3}
0	10	80	30	40	S: {0, 1, 3, 4}
0	10	80	30	40	S: {0, 1, 3, 4, 2}
0	10	80	10	40	

We see that the update with the length of $0 \rightsquigarrow 3$ updates to a lower value but the path $0 \rightsquigarrow 4$ does not update because it is already in S

5. For this problem, we alter Dijkstra's algorithm a little:

Algorithm:

```
d[s] ← 0
e[s] ← 0

for each v ∈ V − {s}
    do d[s] ← ∞
       e[s] ← ∞

S ← ∅
Q ← V

while Q ≠ ∅
    do u ← Extract_Min(Q)
       S ← S ∪ {u}
       for each v ∈ Adj[u]
           do if d[v] > d[u] + w(u,v)
                 then d[v] ← d[u] + w(u,v)
                      e[v] ← e[u] + 1
              else if d[v] == d[u] + w(u,v)
                 then if e[u] + 1 < e[u]
                         then d[v] ← d[u] + w(u,v)
                              e[v] ← e[u] + 1
```