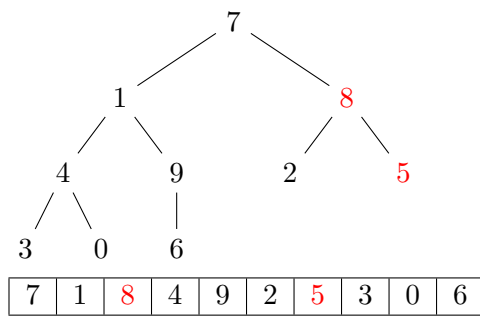
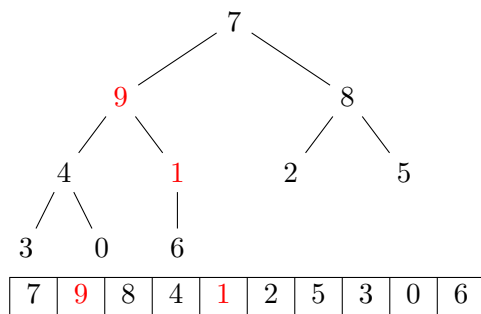


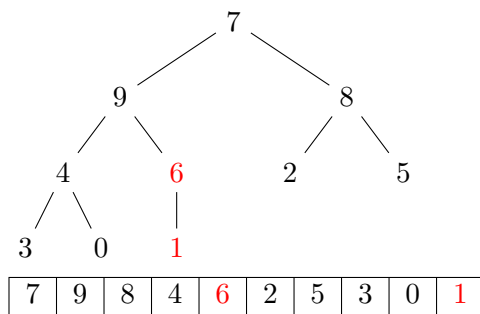
(a) initial state



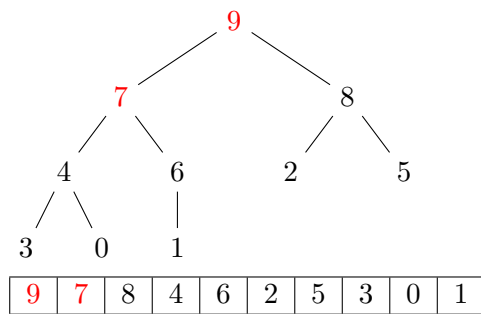
(b) swap 5 and 8



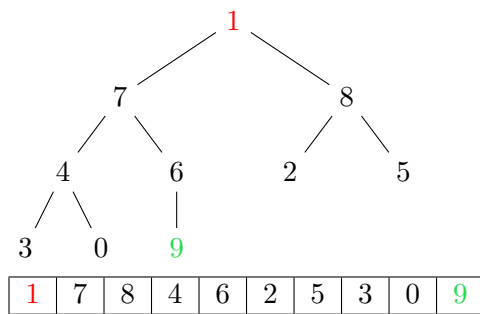
(c) swap 1 and 9



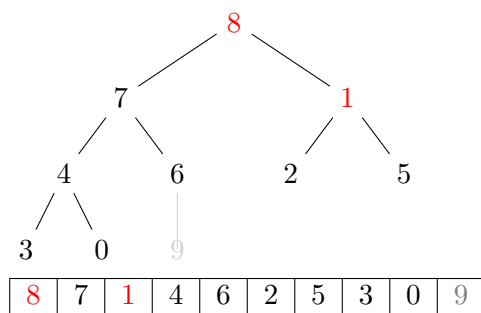
(d) swap 1 and 6



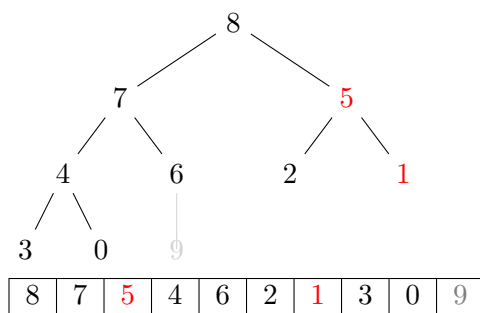
(e) swap 7 and 9



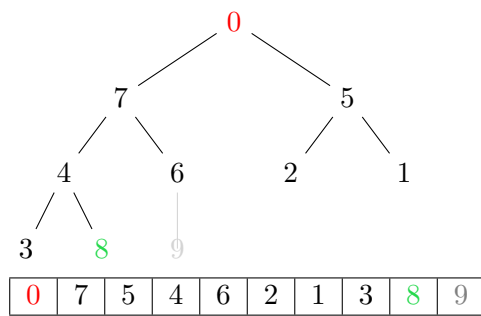
(f) max heap - swap 1 and 9



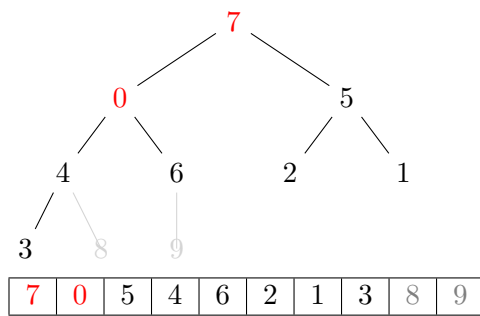
(g) swap 1 and 8



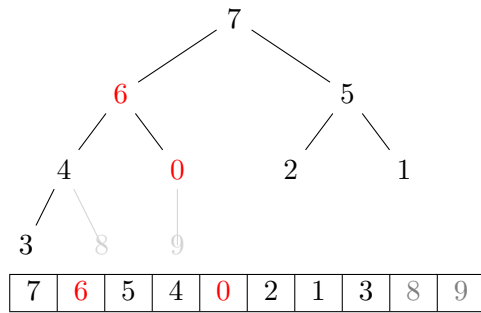
(h) swap 1 and 5



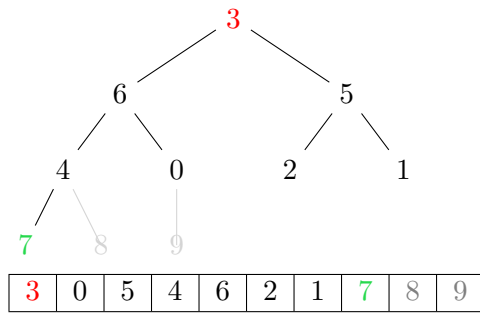
(a) max heap - swap 0 and 8



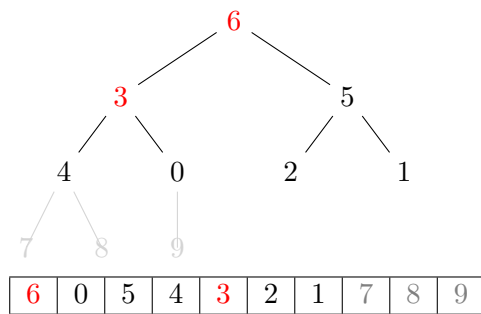
(b) swap 0 and 7



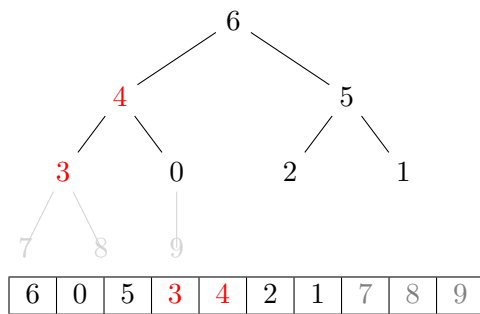
(c) swap 0 and 6



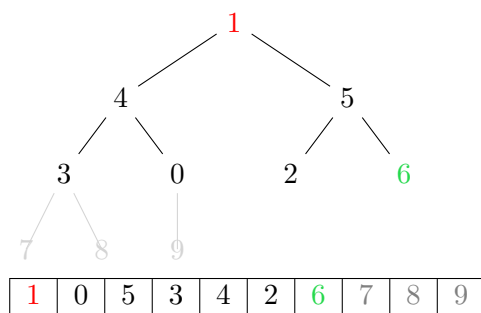
(d) max heap - swap 3 and 7



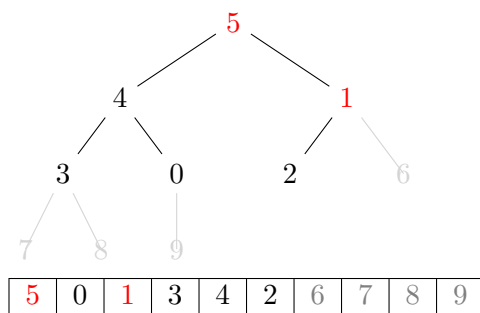
(e) swap 3 and 6



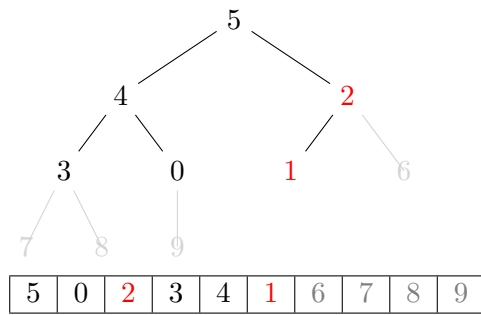
(f) swap 3 and 4



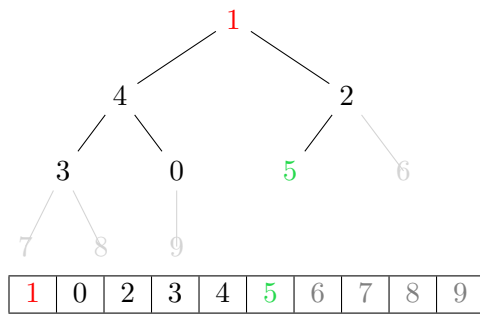
(g) max heap - 1 and 6



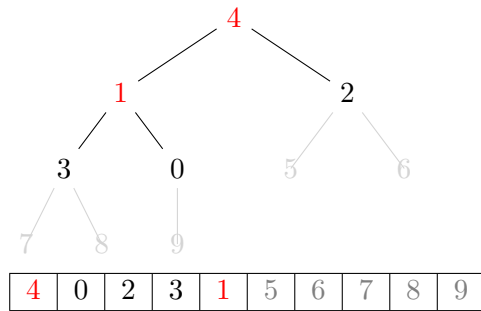
(h) swap 1 and 5



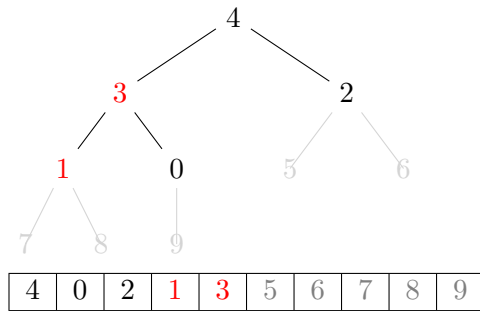
(a) swap 1 and 2



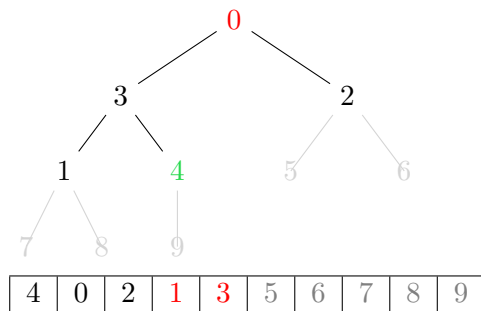
(b) max heap - swap 1 and 5



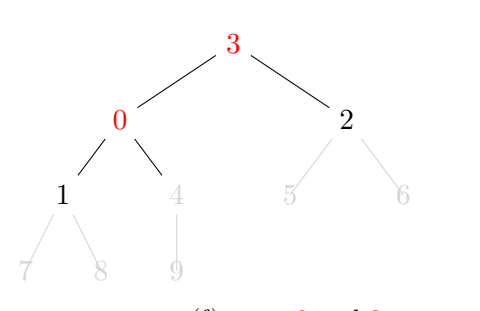
(c) swap 1 and 4



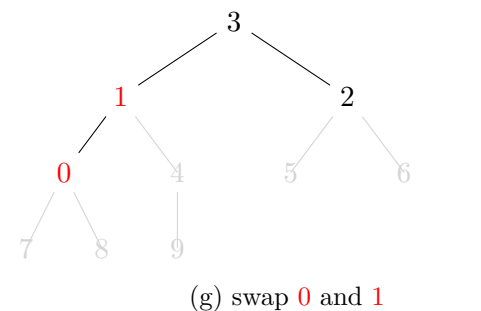
(d) swap 1 and 3



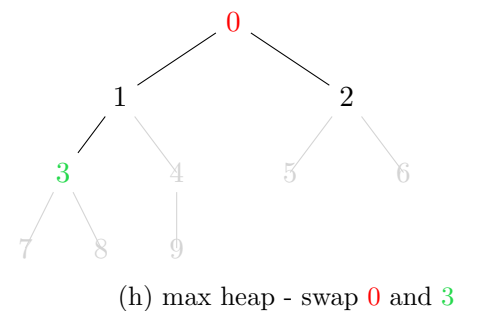
(e) swap 0 and 4



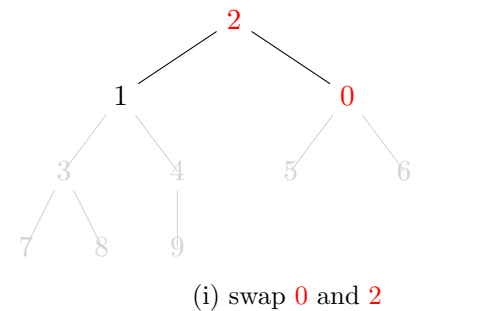
(f) swap 0 and 3



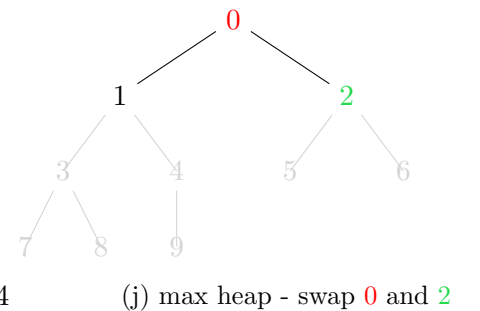
(g) swap 0 and 1



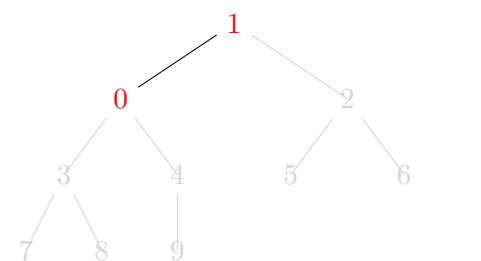
(h) max heap - swap 0 and 3



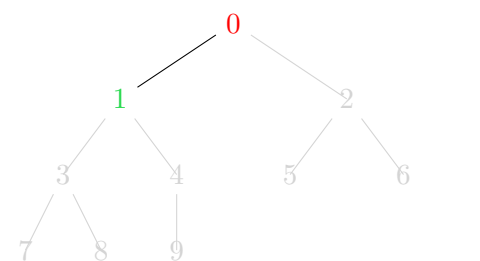
(i) swap 0 and 2



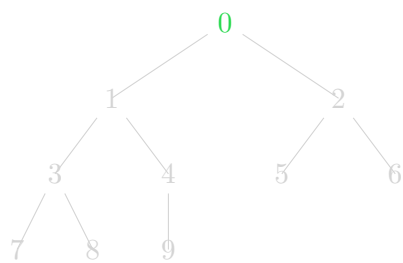
(j) max heap - swap 0 and 2



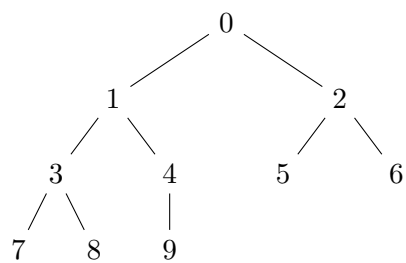
(k) swap 0 and 1



(l) max heap - swap 0 and 1



(a) max heap - swap 0 and 0



(b) the binary tree is correctly sorted

0	1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---	---

(c) deconstruct the binary tree to a sorted list