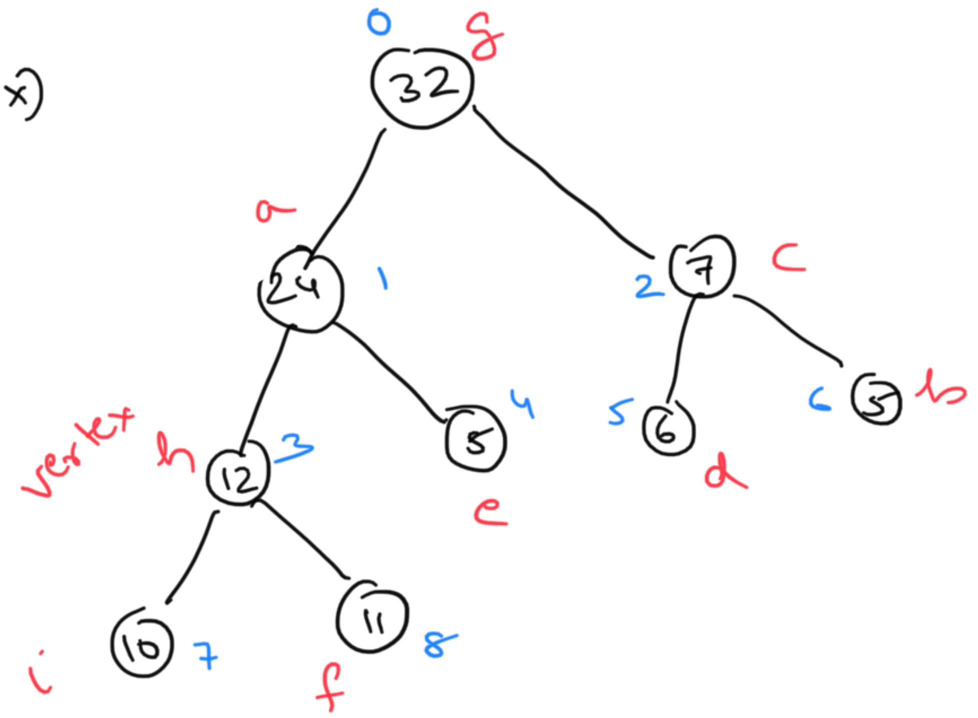
# Heaps again

Sometimes you want to update values in a heap

(ex)



red colour: vertex label corresponding to datum in original problem

blue colour: node location in heap

maintain

(I) vertex to heap node

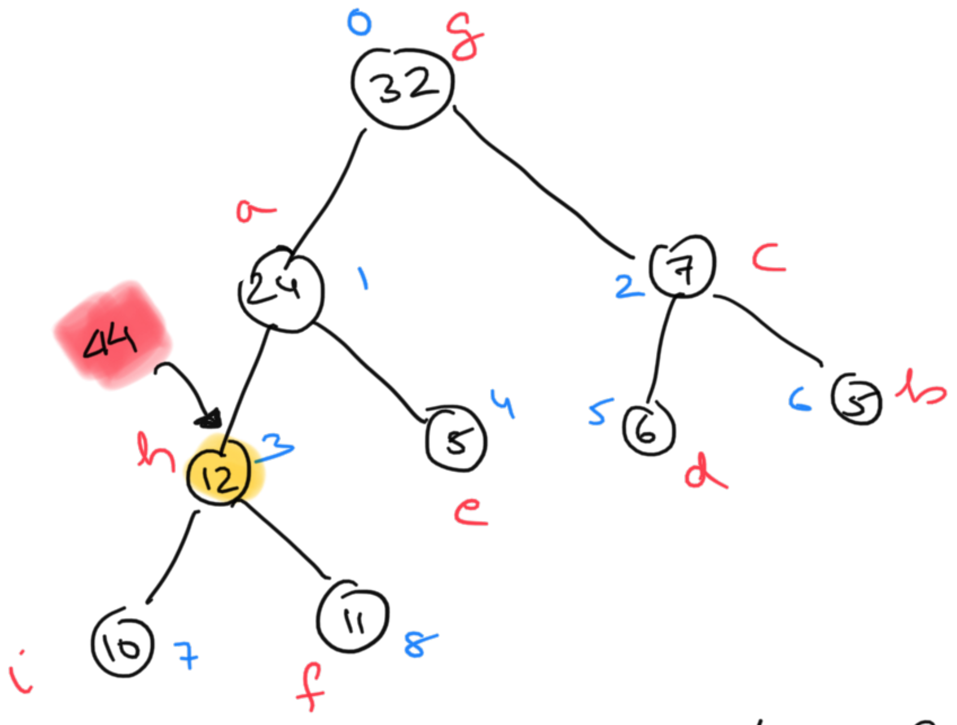
a : 1, b: 6, c : 2, d : 5, e :4, f : 8, g : 0, h: 3, i: 7

(II) heap node to vertex

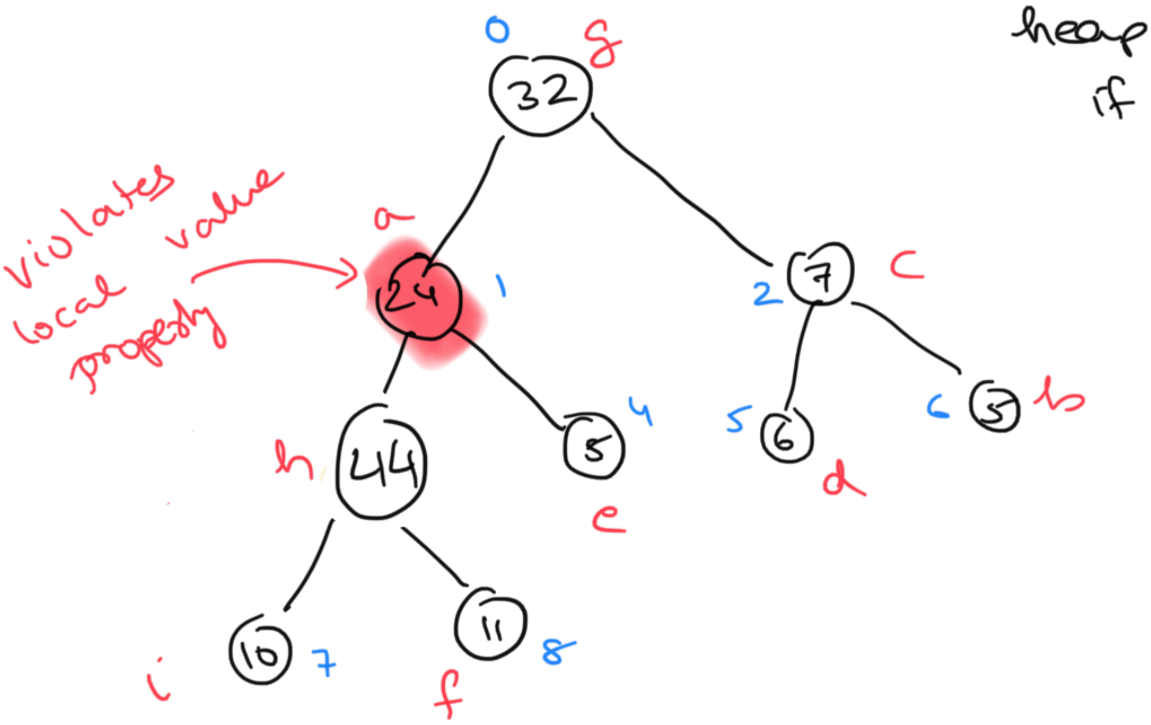
0: g, 1: a, 2: c, 3: h, 4 : e, 5 : d, 6 : b, 7 : i, 8 : f

---

Say you want to update vertex h's value from 12 to 44 in original problem.



44 > 12., so heap property if at all will be violated upwards.


violates local value property

So swap 24 and 44

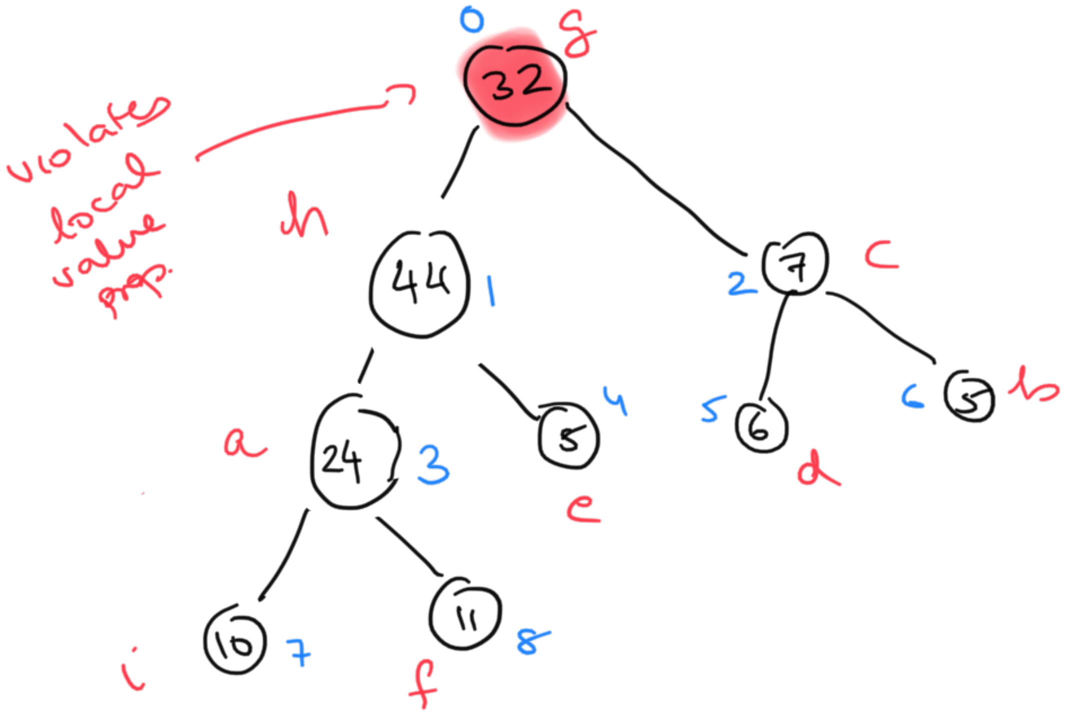
Update

→ I vertex to heap node

a : ~~1~~ 3, b: 6 , c : 2 , d : 5 , e : 4, f : 8 , g : 0, h : ~~3~~ 1, i : 7

→ II heap node to vertex

0 : g , 1 : ~~a~~ h , 2 : c, 3 : ~~h~~ a, 4 : e, 5 : d, 6 : b, 7 : i, 8 : f

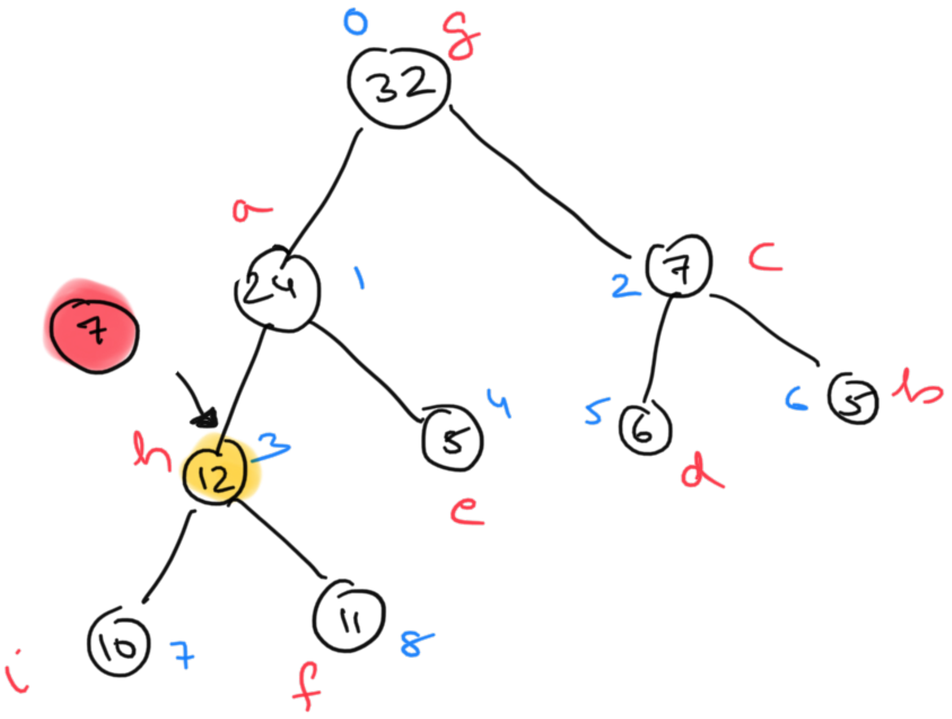


so swap 32 and 44....

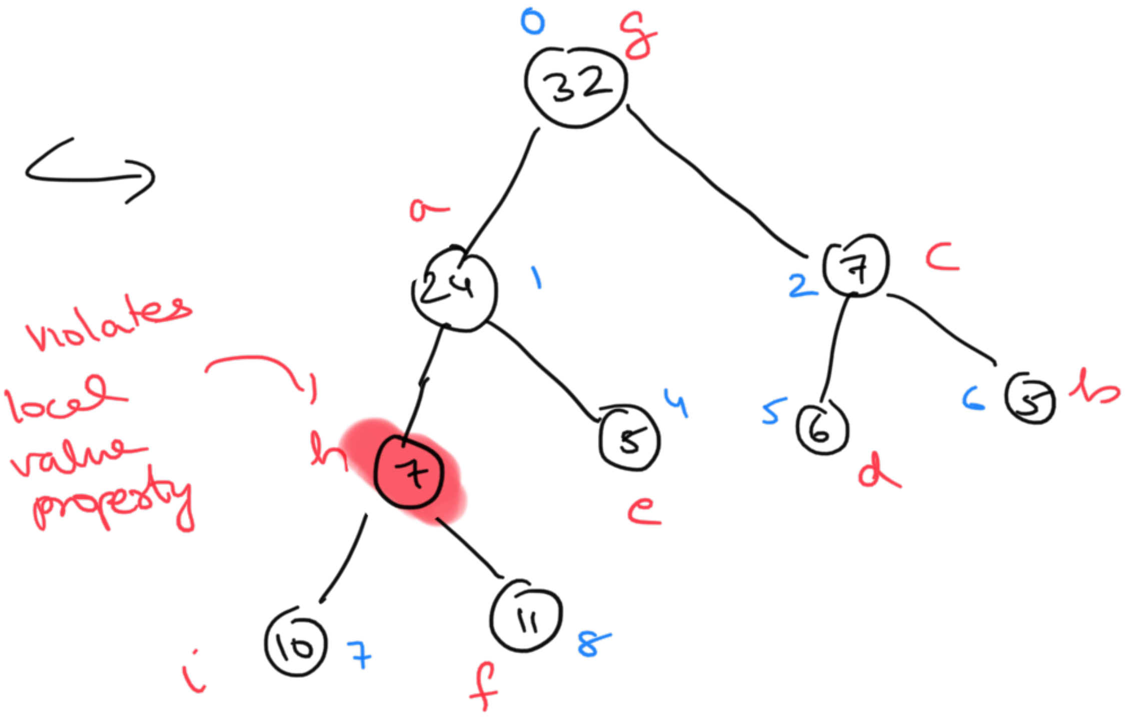
so you end up walking up heap to repair it

---

Note if instead you wanted to replace 12 with 7, ....



you'll walk down the tree repairing heap...



→ so swap 7 with max of its children

max ( 10, 11 ) = 11

→ so swap 7 and 11

⋮

so you end up walking down the heap

---

So updating one value → log n

→ updating all values in heap

O( n log n)