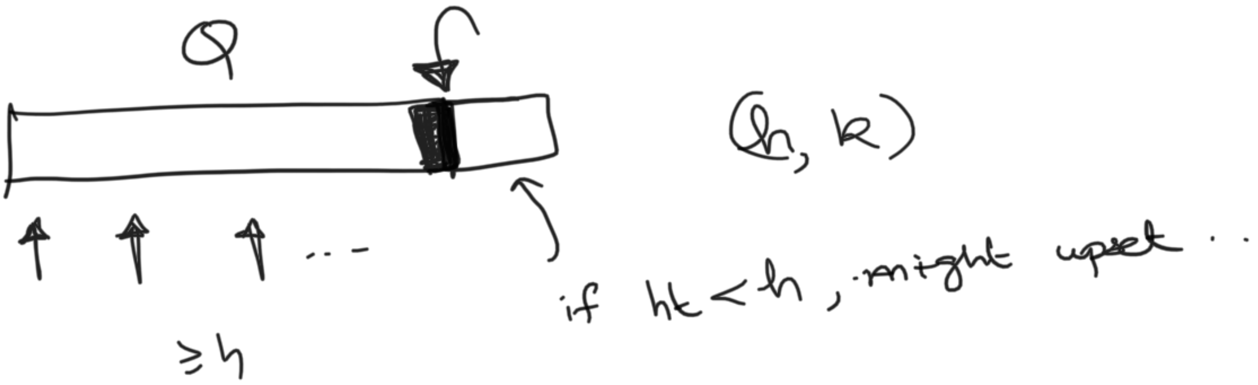
Sorting: Queue reconstruction by height

ht count

given a list of tuples (h, k)

construct a queue where

elt (h, k) ⇒ there are exactly k elements before (h, k) with "ht" ≥ h



so insert in dec order of hts

eg [7,0], [4,4], [7,1], [5,0], [6,1], [5,2]

Sort by dec order of hts and then by inc order of count

[7,0], [7,1], [6,1], [5,0], [5,2], [4,4]

insert in pos

↓

[7,0]

q-length = 1

Insert in 1 pos.

[7,0] [7,1]   q_length= 2

[7,0] [6,1] [7,1]

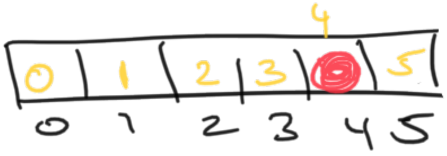
[5,0] [7,0], [6,1], [7,1] ← insert in 0 pos

[5,0], [7,0], [5,2], [6,1], [7,1] ← insert in 2 pos

[5,0], [7,0], [5,2], [6,1], [4,4] [7,1] ← Insert in 4 pos

Insert & Inserting in bet ....

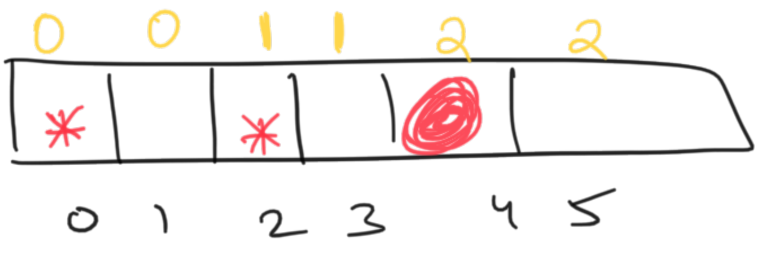
shortest person → is [4,4] has to be in 4th pos..



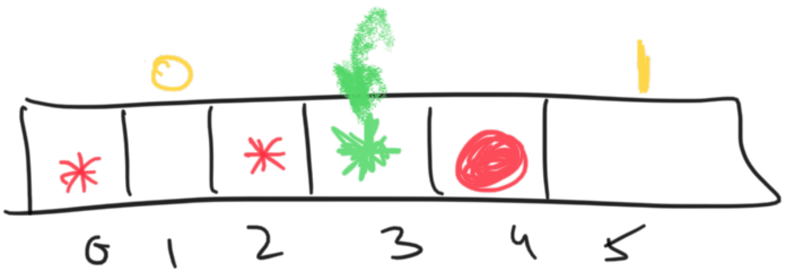
← number of empty slots before each index

next shortest person → is [5,0], [5,2)

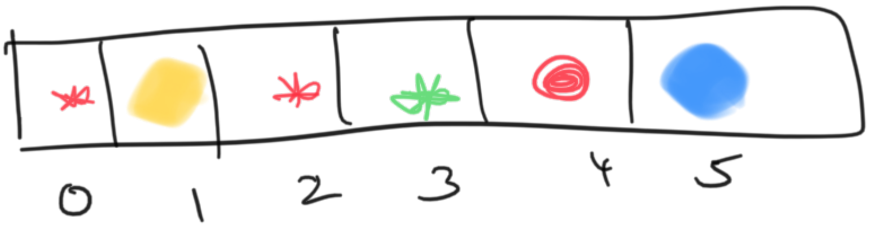
as 0,2 before 4, they are in 0,2 pos



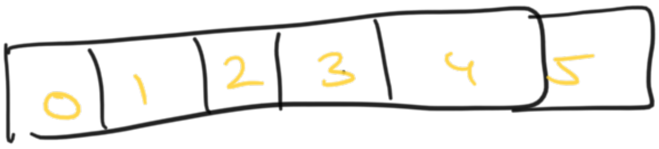
next → [6,1]



next → [7,0], [7,1]



so



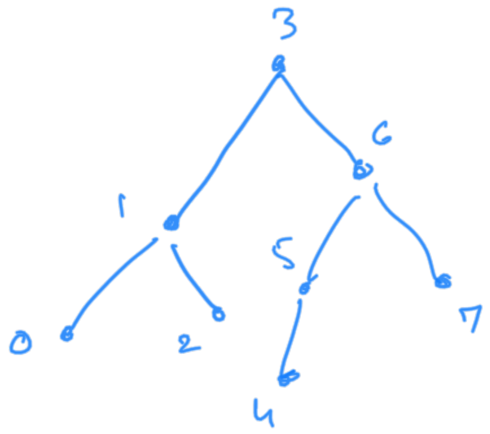
number of empty slots before each index

insert [h,k] at the first index pos which has empty slots = k

for each position > pos, decrement empty slots by 1.

·——

Can use height-balanced trees to make insert operations logarithmic time.



↓ insert after 5 & before 6