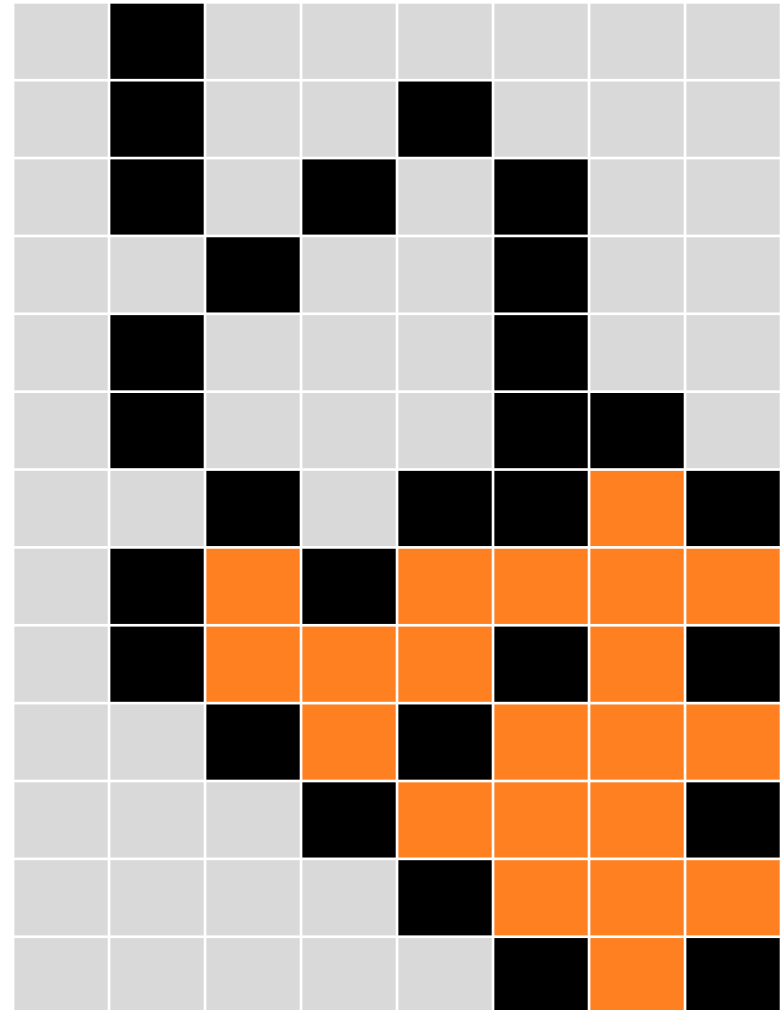
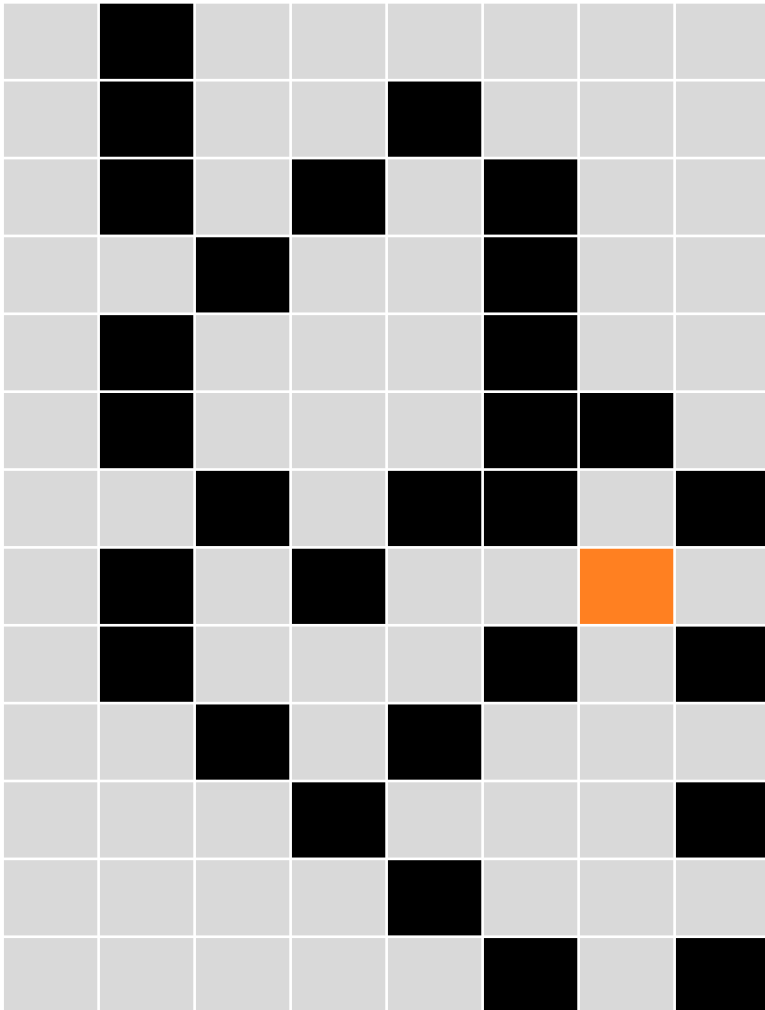


Review on Previous Lectures

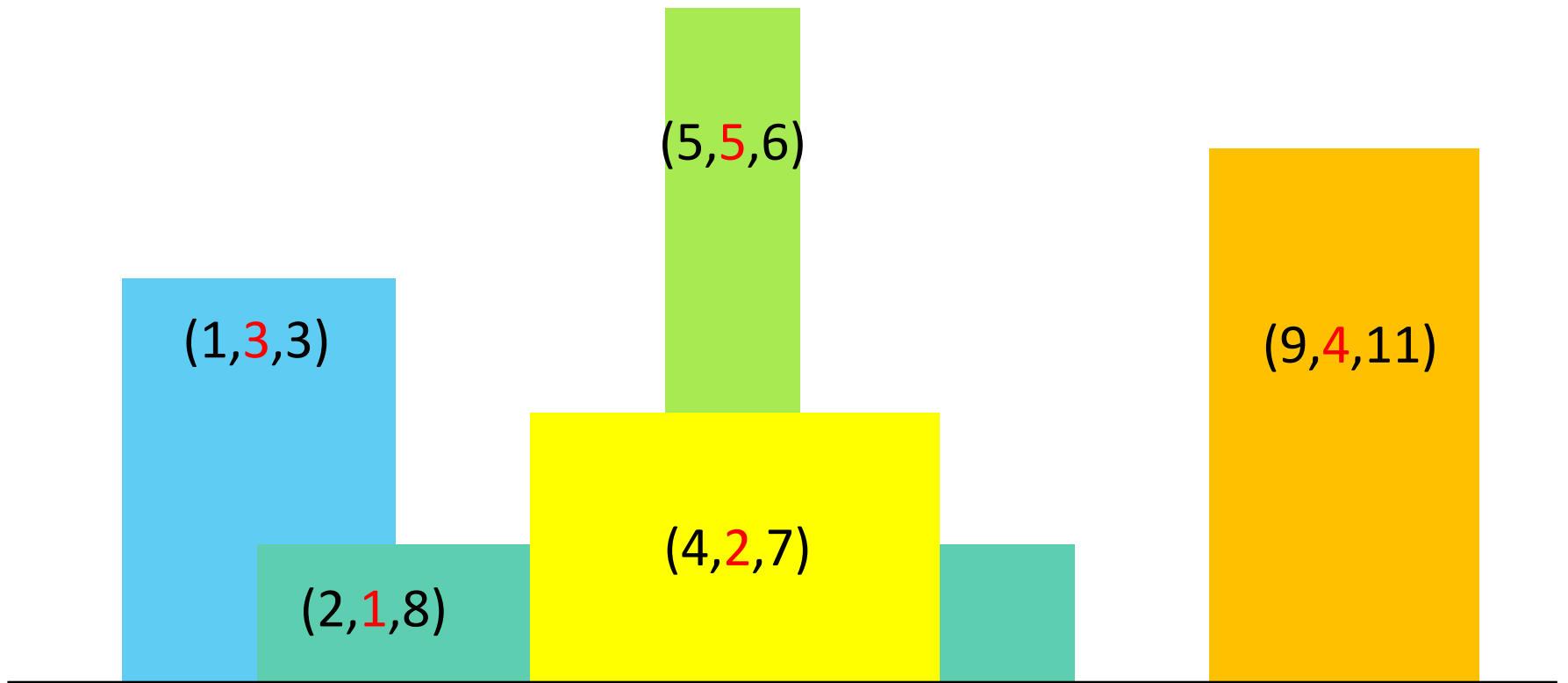
Flood fill



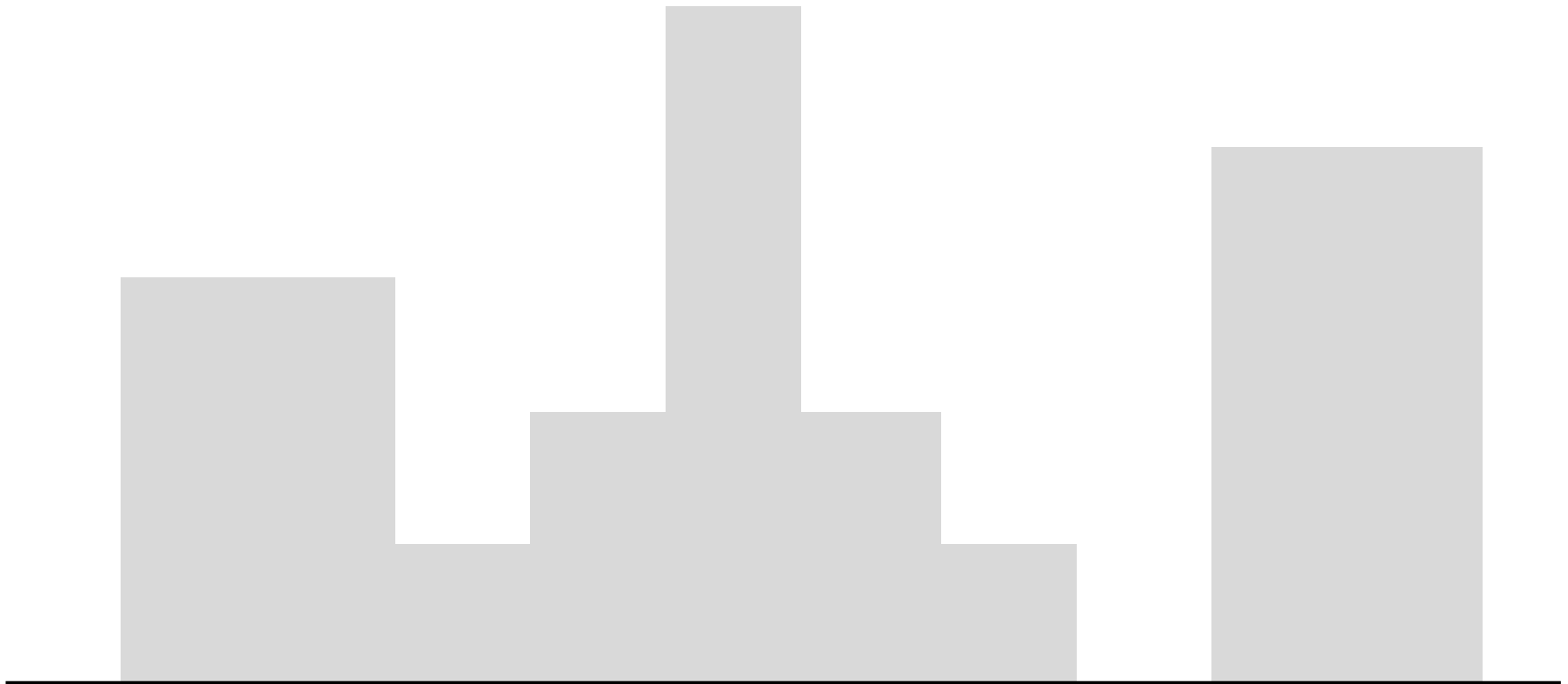
Flood fill

- How to implement?
- Queue
 - Put the source into queue
 - While the queue is not empty, color the first inserted element in the queue, then remove it and put its colorable adjacent cells into queue.
- Recursive calls
 - Color the source.
 - If an adjacent cell X is colorable, then using X as the source to recursively call the flood fill function.

Skyline



Skyline

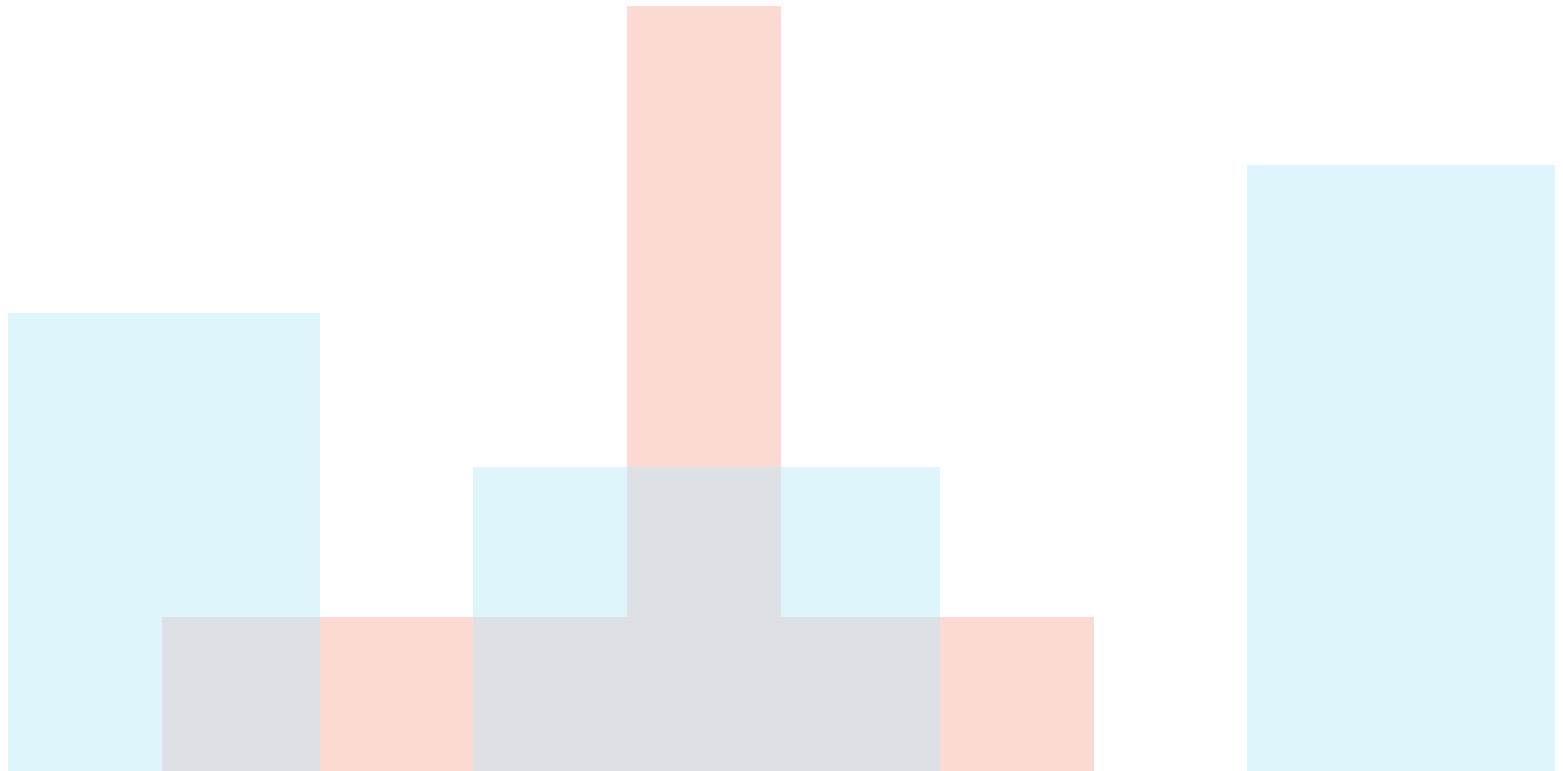


(1,3,3,1,4,2,5,5,6,2,7,1,8,0,9,4,11)

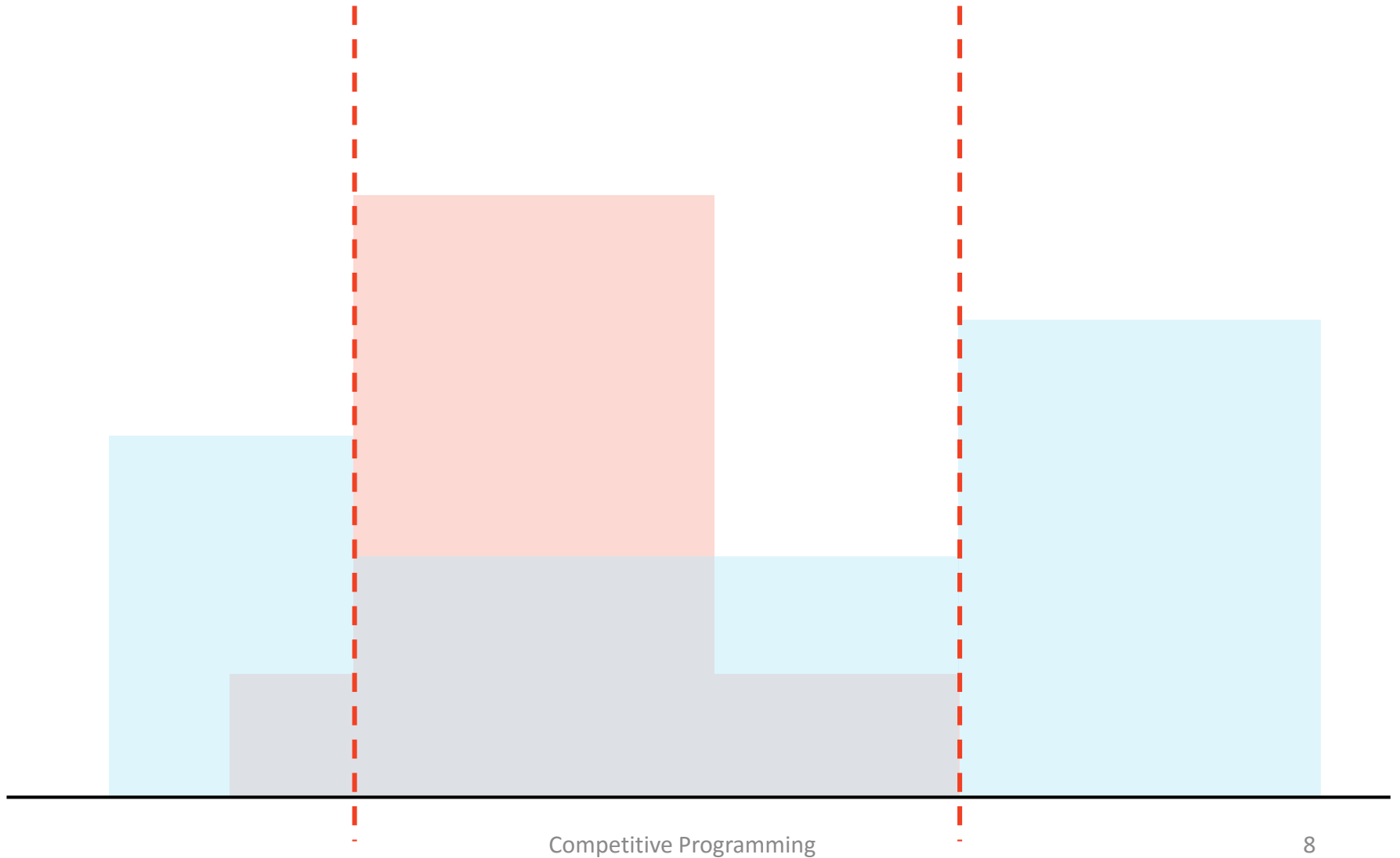
Skyline in $O(n \log n)$

- Divide and Conquer
 - A skyline can be represented as a sequence $(x_1, y_1, x_2, y_2, x_3, y_3, \dots, x_{n-1}, y_{n-1}, x_n)$
 - Terminal condition: a building (x_1, y_1, x_2)
 - Merge two skylines in $O(n)$ -time.
- Sort + Priority Queue
 - All changes must occur at boundaries of buildings.
 - A new tallest building is inserted
 - The tallest building is removed
 - Delay the removes

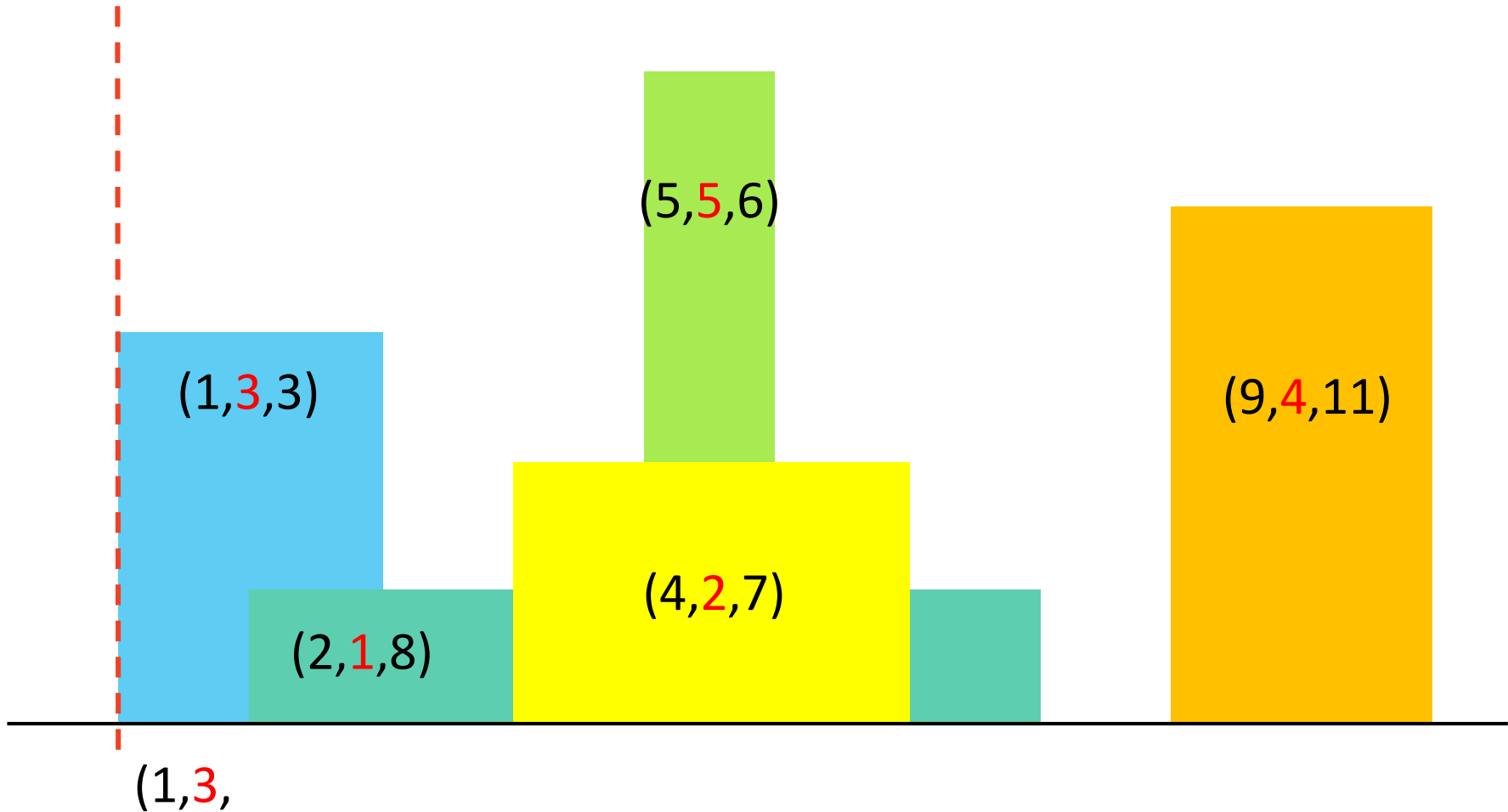
(1,3,3,0,4,2,7,0,9,4,11)
+ (2,1,5,5,6,1,8)
= (1,3,3,1,4,2,5,5,6,2,7,1,8,0,9,4,11)



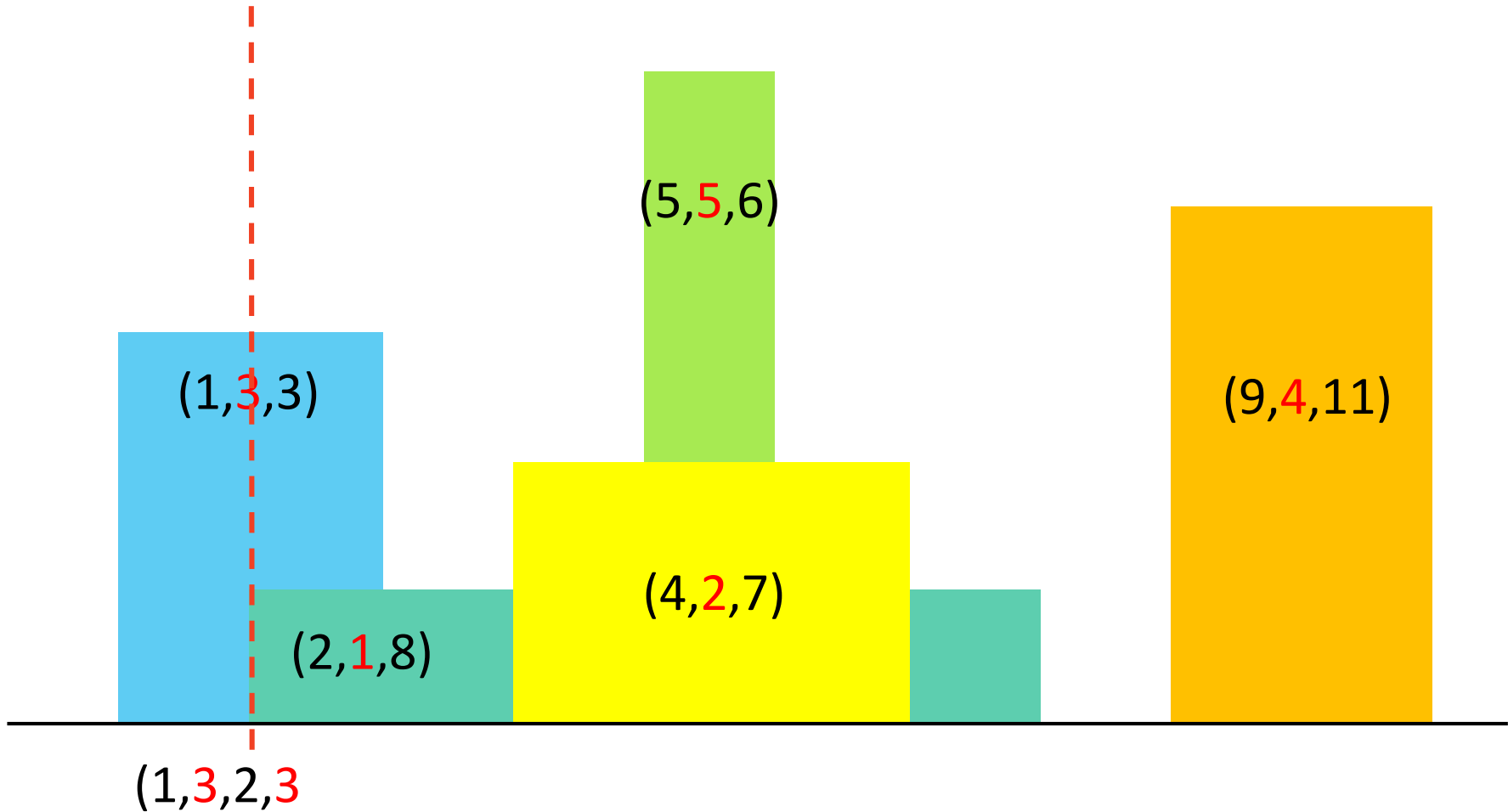
Trouble maker



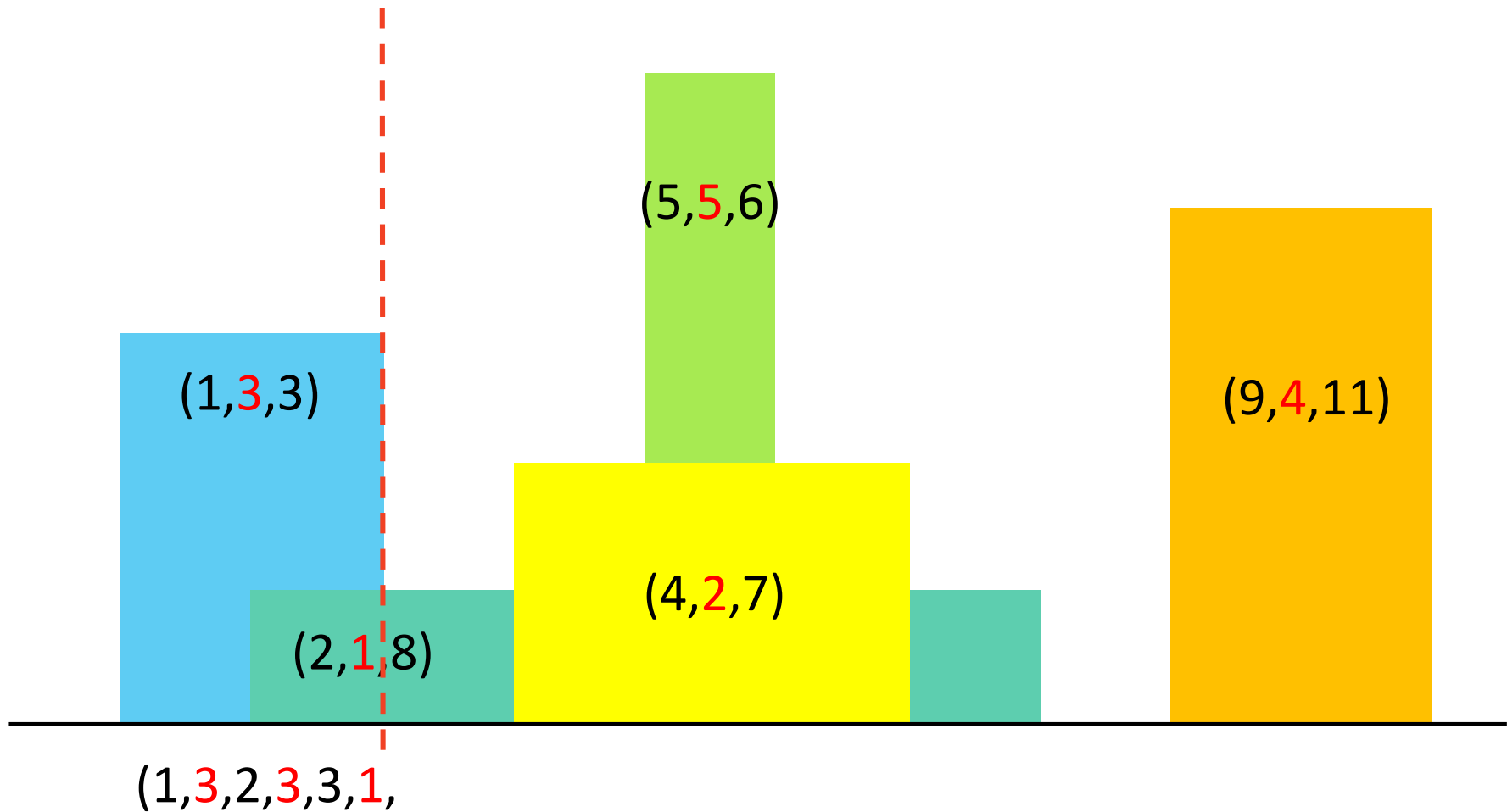
Priority queue: $\{(1, \textcolor{red}{3}, 3)\}$



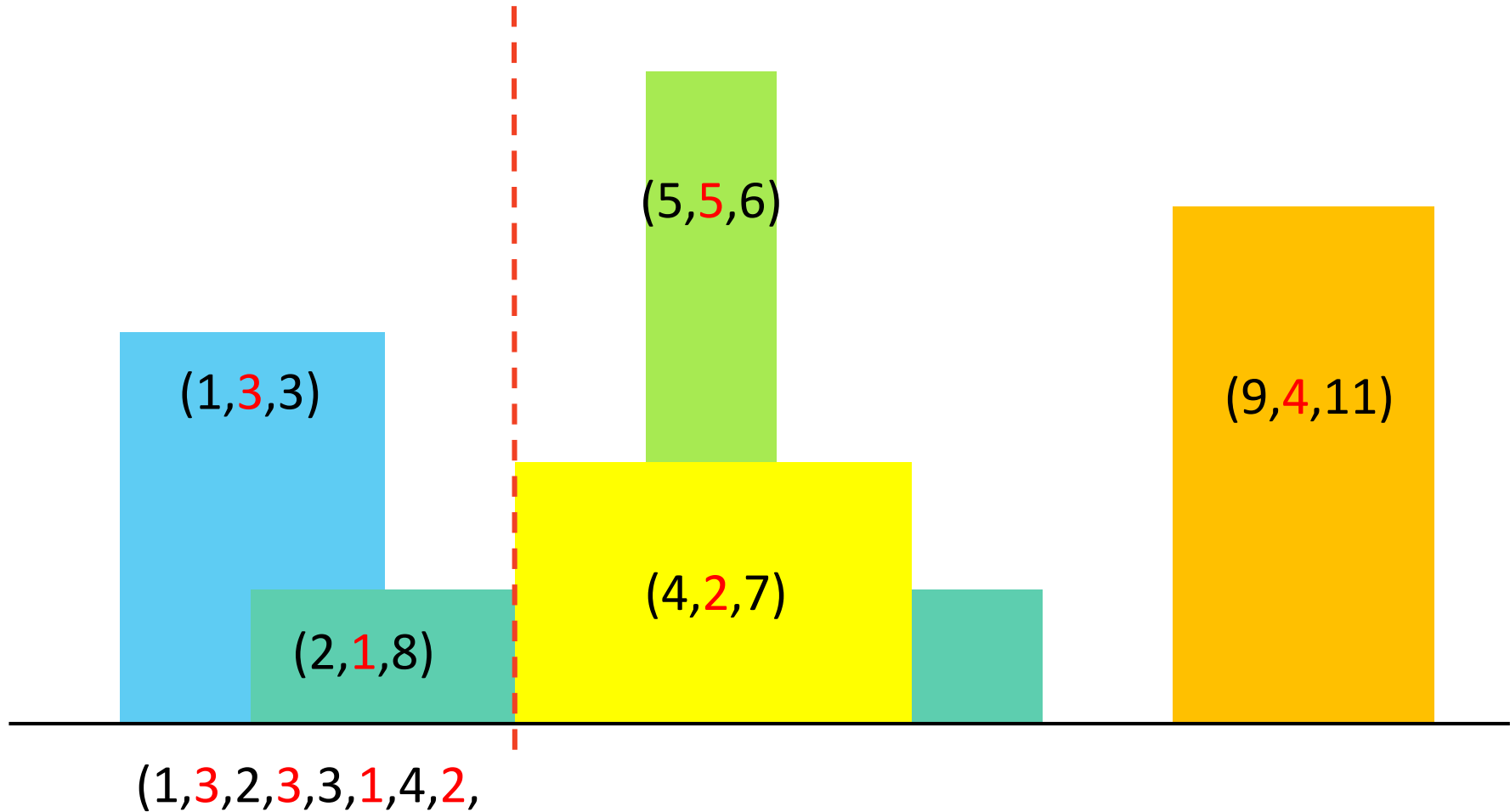
Priority queue: $\{(1, \textcolor{red}{3}, 3), (2, \textcolor{red}{1}, 8)\}$



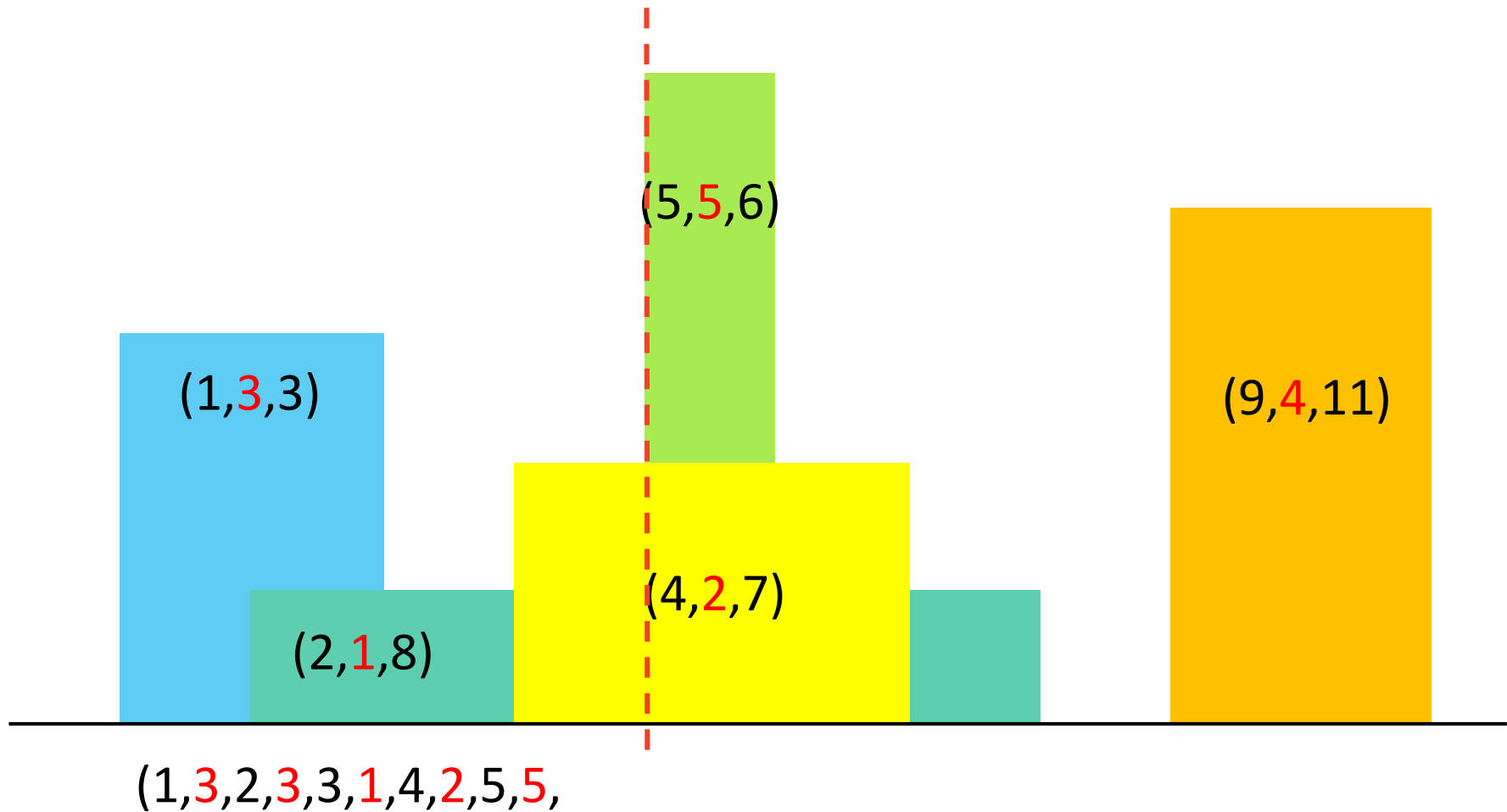
Priority queue: $\{(2,1,8)\}$



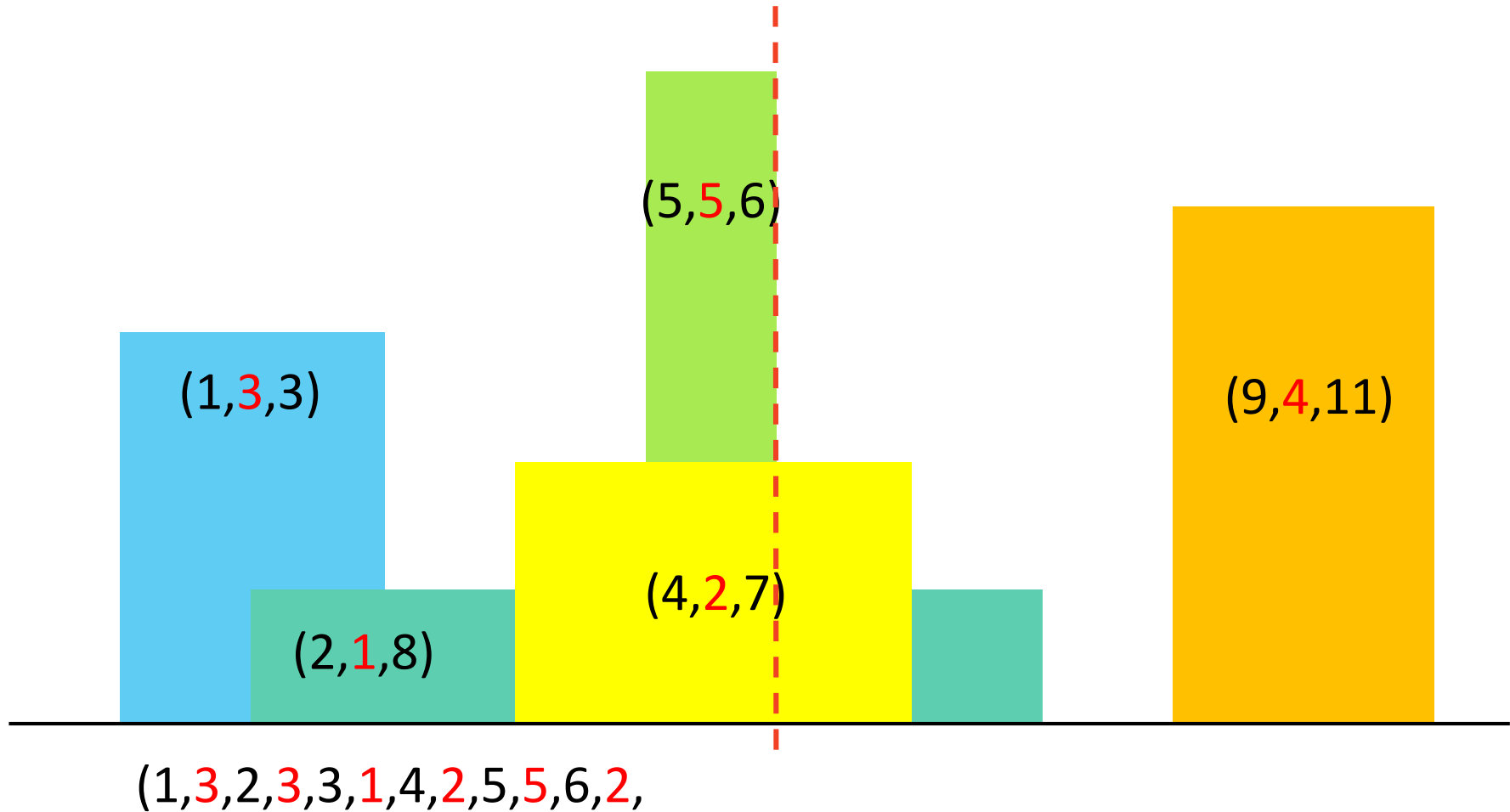
Priority queue: $\{(4,2,7),(2,1,8)\}$



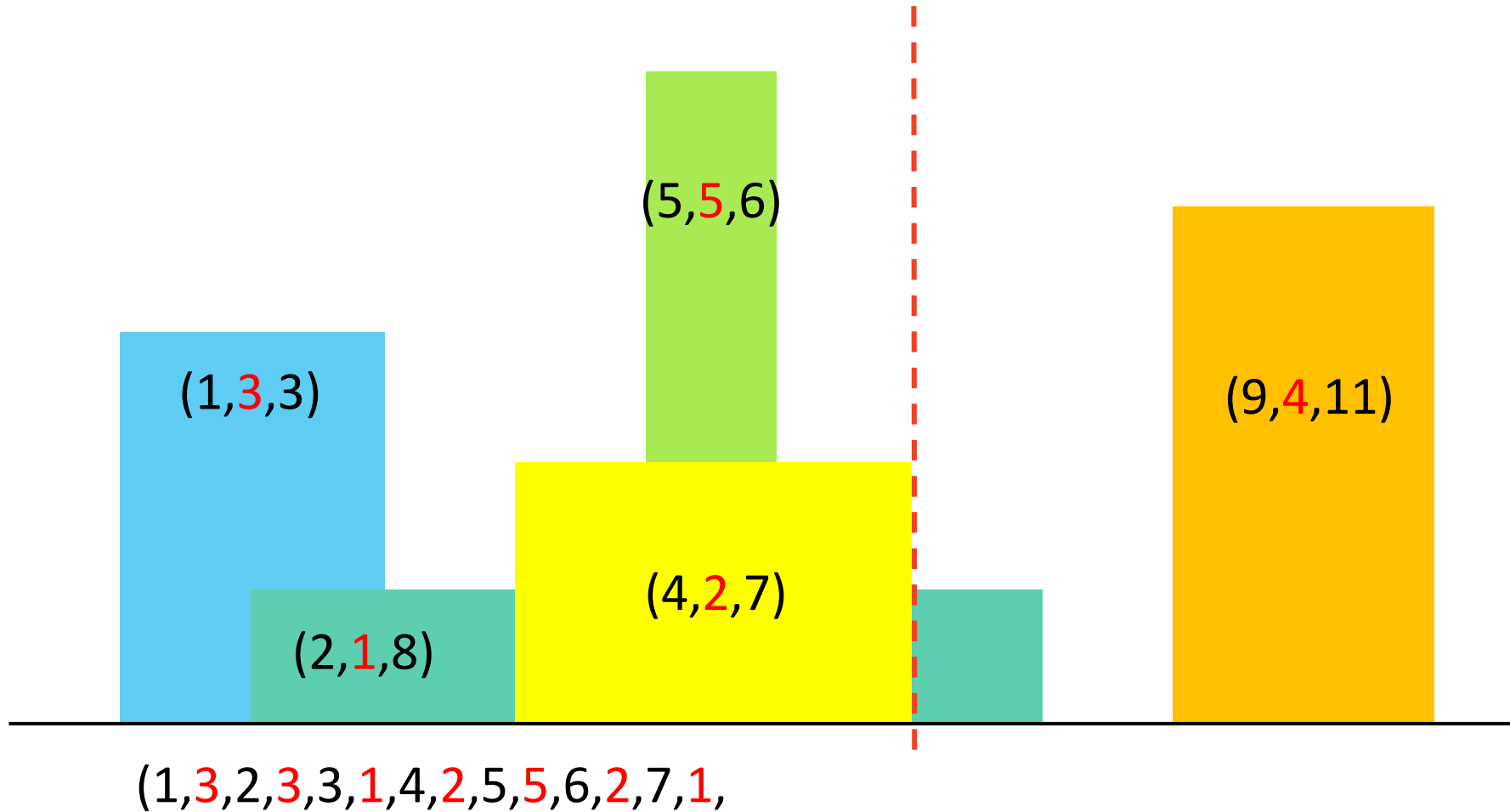
Priority queue: $\{(5,5,6),(4,2,7),(2,1,8)\}$



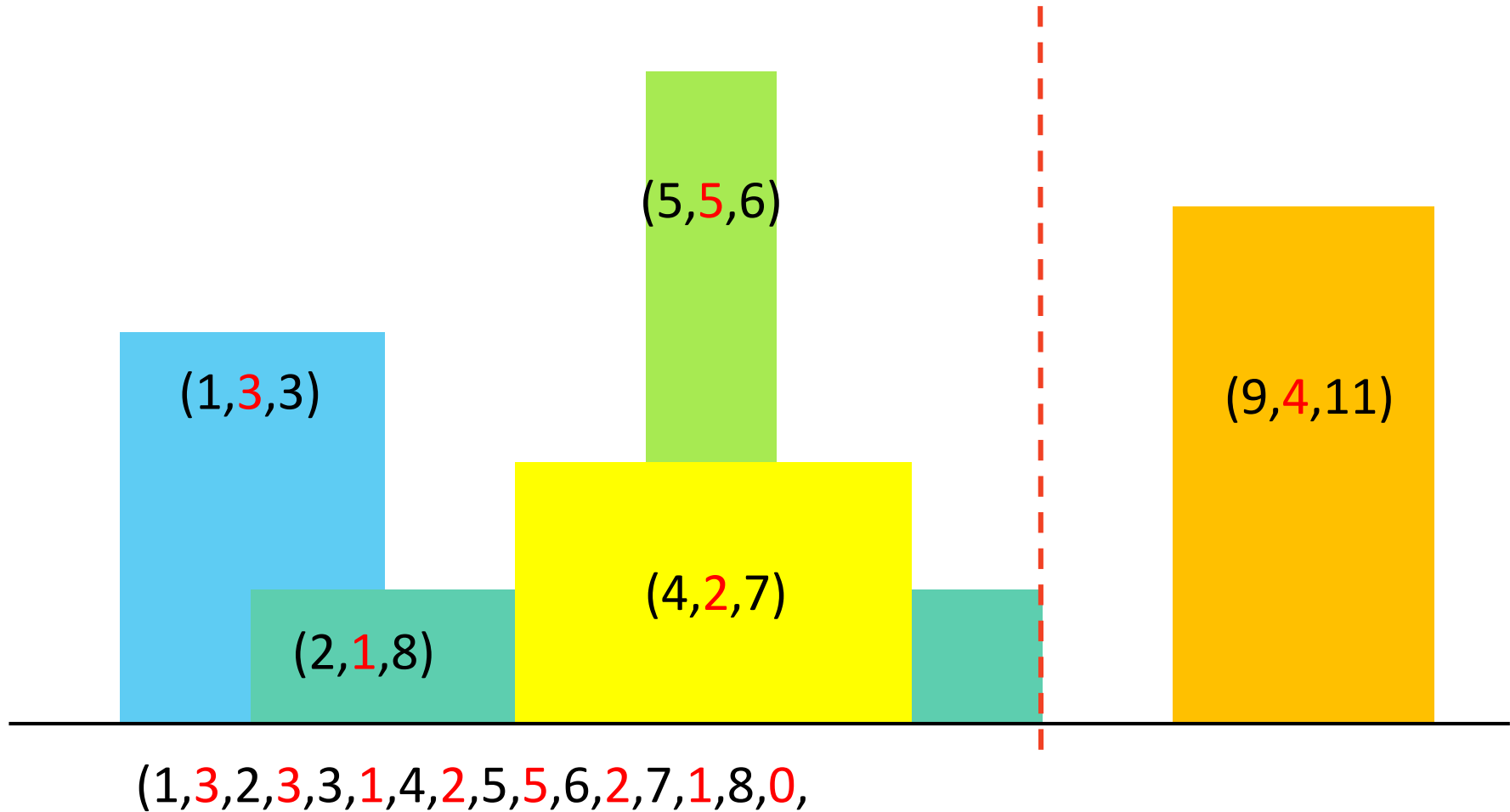
Priority queue: $\{(4,2,7),(2,1,8)\}$



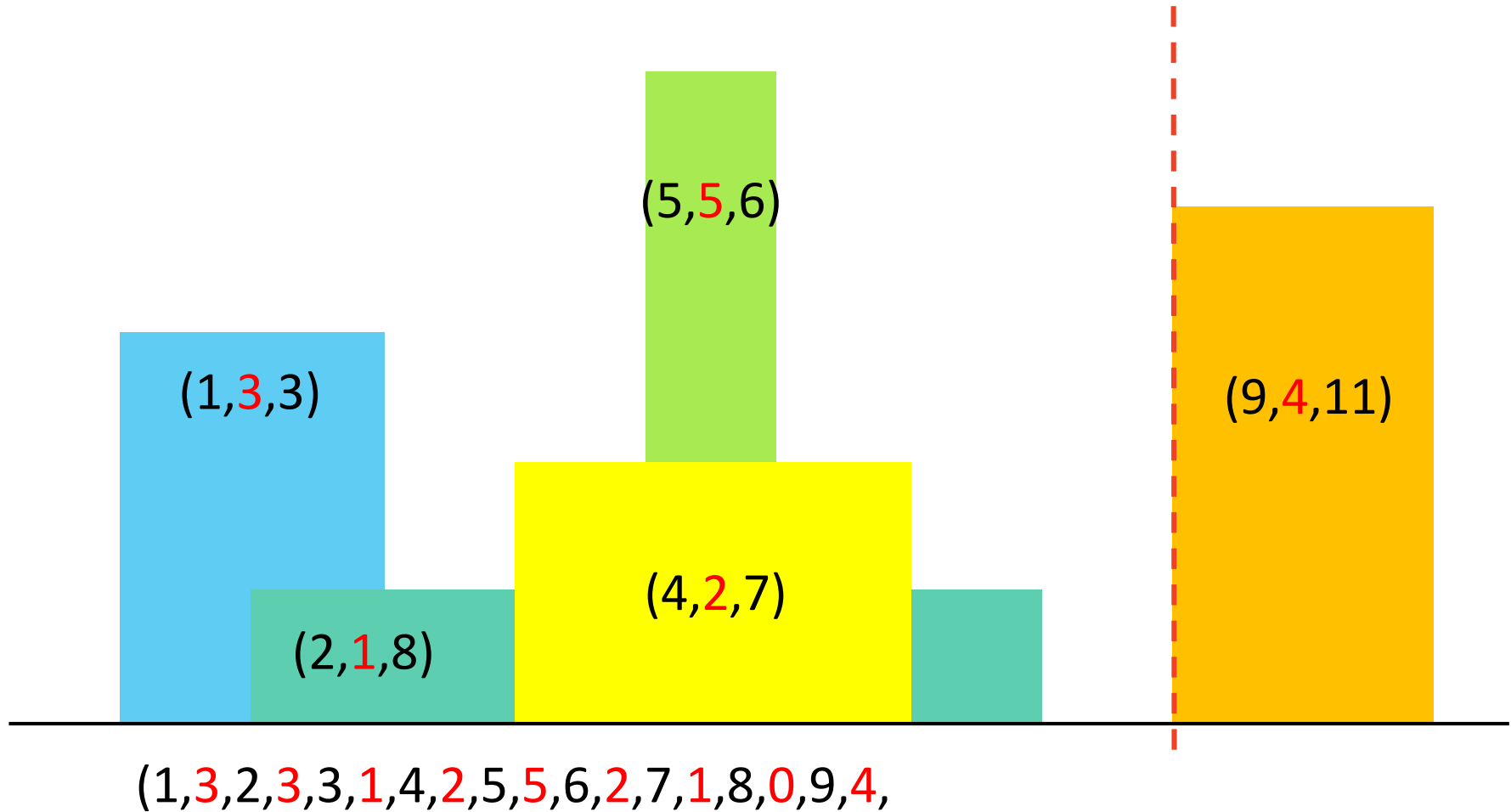
Priority queue: $\{(2,1,8)\}$



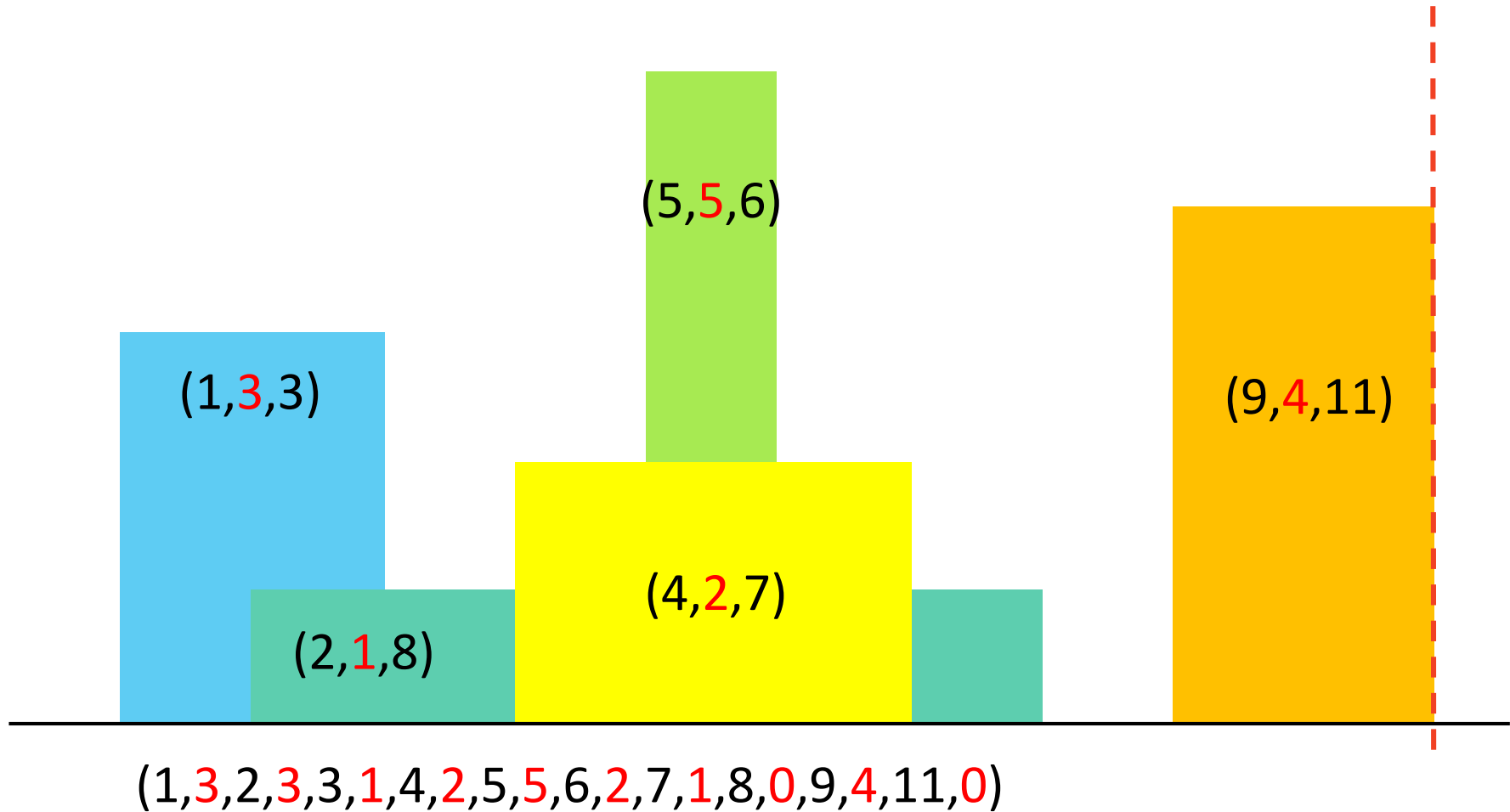
Priority queue: {}



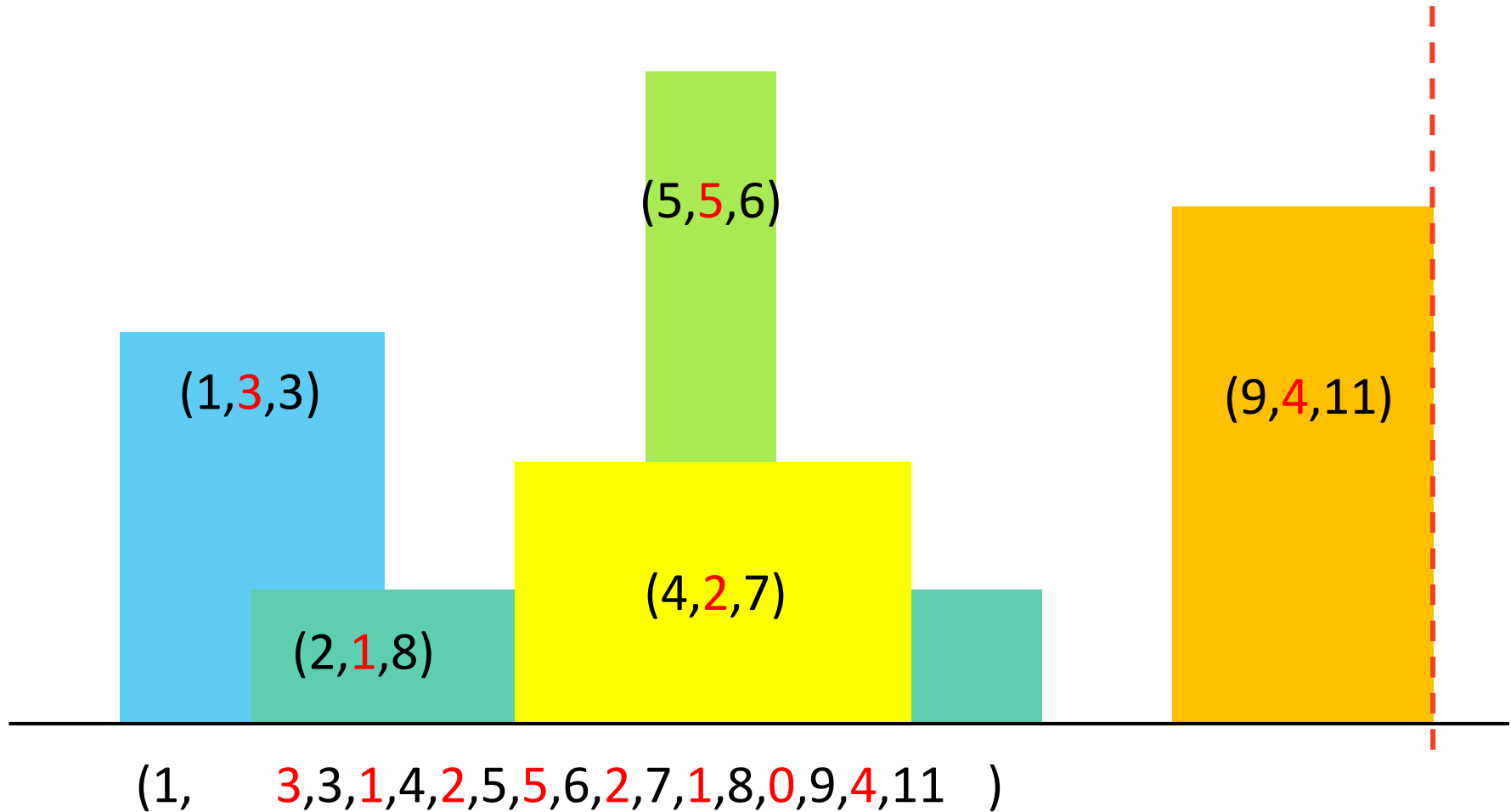
Priority queue: {(9,4,11)}



Priority queue: {}



Priority queue: {}



Trouble maker

