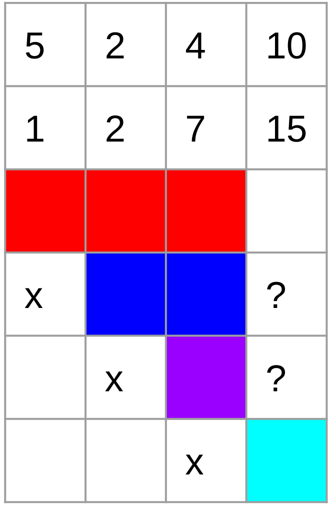
P2. Hospital Construction

[[홍원표]]



5	2	4	10
1	2	7	15
🟥	🟥	🟥	
x	🟦	🟦	?
	x	🟪	?
		x	▨

Time Complexity: O(nlogn + n^2)
Space Complexity: O(4n + c)

Since this question is not about finding WHERE the hospital should be built but the HIGHEST number of people, there is NO NEED to build an array for ALL points in between.
However, since this problem is all about distances, sorting is 100% necessary. [nlogn]

The leftmost index (smallest ‘point’) will not have any point on its left side.
This means its MAX distance range will FROM itself TO itself+2k.

In this example, since the leftmost index starts at ‘point’ 1, the range for that index will be FROM 1 TO 7. (Highlighted with RED)

The next index does NOT need to include the previous index because its MAX has already been found. In another words, the next index ALSO should count FROM itself TO itself+2k. This process can be simplified even more by obtaining the result of the previous as a scratch(excluding the previous index) and start from there.
Its time complexity, at most, will be searching until the index finishes, so it would be (n-1)(n-1) times.

In this example, BLUE part is for the next index. It excluded the previous index (shown as ‘x’), contains the rest of the sum, and checks for the next index (shown as ?). Since 15 > 7+3+3, it doesn’t include.

This algorithm can be applied to any real-life problem that has its environment in a straight line.
For example, searching for the right place for
a convenience store (In order to attract the most subway-commuters)
in a subway station (subway runs through a straight line back-and-forth)
would be excellent example to solve with this algorithm.