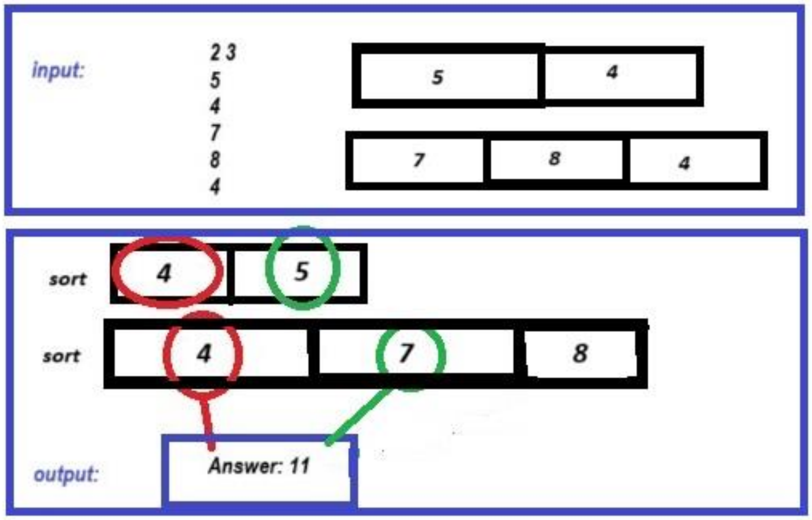
a. Identify the greedy choice:
Select the first knight that is tall enough to chop off the current dragon head from a non-decreasing sorted list.

b. Identify the optimal substructure:
OptimalSolution(Global) = Knight[n] >= DragonHead[n] ? (LocalSolution) : Knight[n+1]

Choose the knight who can chop off the current dragon head with the minimum cost. At making this choices we do LocalSolutions and contributes to the global optimal solution.

c. Proof of the greedy algorithm:

The greedy algorithm works by always selecting the knight who can chop off the current dragon head with the minimum cost. This ensures that we minimize the total cost of hiring knights while still being able to defeat the dragon.



d. Time complexity of the solution:

N log N