

C Problem Analysis - Manuel MOrales

Greedy Choice:

- At each step, choose the knight with the minimum height who can chop off the current dragon head.

Optimal Substructure:

- By choosing the knight with the minimum height who can chop off the current dragon head, we ensure that the remaining knights are available for future dragon heads. This optimal substructure allows us to make the greedy choice at each step.

Proof of Greedy Algorithm:

- **Greedy Stays Ahead:** We can prove the correctness of the greedy algorithm by showing that at each step, the greedy solution is at least as good as the optimal solution. Let G be the solution obtained by the greedy algorithm and O be the optimal solution. We define a measure as the total height of knights used to defeat dragon heads. We can prove that for each dragon head, the height of the chosen knight by the greedy algorithm is always greater than or equal to the height of the knight chosen by the optimal algorithm. Therefore, the greedy solution always stays ahead in terms of the total height used, ensuring that it produces the optimal solution.

Time Complexity:

- The time complexity of the solution is $O(n \log n + m \log m)$, where n is the number of dragon heads and m is the number of knights. This complexity arises from sorting the dragon heads and knights arrays.