730

Intuition:

At any point you can only start projects whose capital requirement ≤ current capital. Among those available, to maximize capital growth fastest, you should pick the one with the largest profit.

So you need a fast way to:

1. Discover which projects have become available as your capital grows.
2. From available ones, pick the maximum profit.

Data structures that make it easy:

- Sort all projects by capital [i] ascending (keep pairs (capital, profit)).
- Maintain a max-heap by profit for projects that are currently available.

Greedy loop (do at most k times):

1. While the next project in the capital-sorted list has capital ≤ current-w, push its profit into the max-heap.
2. If the heap is empty → no project is affordable → stop early (you can't grow capital).
3. Pop the largest profit from the heap and add it to your capital.
4. Repeat.

Stop after k picks or when no more are affordable.

Why this works (sketch):

- At each step, your only choice is among currently affordable projects. Picking the highest profit among them maximizes the next capital and never hurts future choices (it only expands affordability faster). This is a matroid/greedy exchange style argument: any optimal schedule can be transformed to one that uses the highest-profit affordable choice at the earliest time without decreasing final capital.

Complexity:

- Sorting: $O(n \log n)$
- Each project pushed/popped at most once: $O(n \log n)$
- Loop skips to k iterations (bounded by n): overall still $O(n \log n)$.