3075. Maximize Happiness of Selected Children

You are given an array happiness of length n, and a positive integer k.

There are n children standing in a queue, where the i^{th} child has happiness value happiness[i]. You want to select k children from these n children in k turns.

In each turn, when you select a child, the happiness value of all children that have not been selected till now decreases by 1. Note that the happiness value CANNOT become negative and gets decremented only if it is positive.

Return the maximum sum of the happiness values of the selected children you can achieve by selecting k children.

Constraints:
- $1 <= n == happiness.length <= 2 \cdot 10^5$
- $1 <= happiness[i] <= 10^8$
- $1 <= k <= n$

My first instinct is just to sort the children in ascending order and popping them from the right of the list until k is zero. My code works as expected, but what could be an edge case that could throw me off? I know that k cannot be larger than n, so popping from an empty list seems to be impossible. k cannot be smaller than 1, so my while loop will always end up in some given time. I am initializing total on the outside of the loop, so it must work. The only thing that I think could cause troubles is the runtime, but lets try it.