You have a certain number of processors, each having 4 cores. The number of tasks to be executed is four times the number of processors. Each task must be assigned to a unique core, and each core can only be used once.

You are given an array processorTime representing the time each processor becomes available and an array tasks representing how long each task takes to complete. Return the *minimum* time needed to complete all tasks.

**Example 1:**

**Input:** processorTime = [8,10], tasks = [2,2,3,1,8,7,4,5]

**Output:** 16

**Explanation:**

Assign the tasks at indices 4, 5, 6, 7 to the first processor which becomes available at time = 8, and the tasks at indices 0, 1, 2, 3 to the second processor which becomes available at time = 10.

The time taken by the first processor to finish the execution of all tasks is max(8 + 8, 8 + 7, 8 + 4, 8 + 5) = 16.

The time taken by the second processor to finish the execution of all tasks is max(10 + 2, 10 + 2, 10 + 3, 10 + 1) = 13.

**Example 2:**

**Input:** processorTime = [10,20], tasks = [2,3,1,2,5,8,4,3]

**Output:** 23

**Explanation:**

Assign the tasks at indices 1, 4, 5, 6 to the first processor and the others to the second processor.

The time taken by the first processor to finish the execution of all tasks is max(10 + 3, 10 + 5, 10 + 8, 10 + 4) = 18.

The time taken by the second processor to finish the execution of all tasks is max(20 + 2, 20 + 1, 20 + 2, 20 + 3) = 23.

**Constraints:**

* 1 <= n == processorTime.length <= 25000
* 1 <= tasks.length <= 105
* 0 <= processorTime[i] <= 109
* 1 <= tasks[i] <= 109
* tasks.length == 4 \* n