Firstly, we store the penalty and time of all assignments in a link list, and then we sort them in descending order according to the ratio of penalty/time, and then enter the loop to accumulate the penalty of each assignment to get the minimise the total number of points lost.

Justify the correctness of your algorithm:

Assuming 2 assignments a(i) and a(j), there are two possibilities for their sorting. The first is to complete a(i) first, and the second is complete a(j) first.

When a(i) first:

Penalty(i) = p(i) \* t(i) + p(j) \* (t(i) + t(j))

When a(j) first:

Penalty(j) = p(j) \* t(j) + p(i) \* (t(i) + t(j))

We assume that Penalty(i) is greater than Penalty(j), then we can get the p(j)\*t(i) > p(i)\*t(j), finally we can get that p(j)/t(j) > p(i)/t(i).

From this relationship, we can know that the greater the ratio of penalty to time, the smaller the penalty we get, so we sort all assignments in descending order of p(i)/t(i).

Time complexity:

In the above algorithm, we use merge sort to sort the link list, the time complexity is O(nlogn), then the penalty is accumulated, its complexity is O(n). Therefore, the time complexity of the whole process is O(nlogn)