2. Your friend is working as a camp counselor, and he is in charge of organizing activities for a set of campers. One of his plans is the following mini-triathlon exercise: each contestant must swim 20 laps of a pool, then bike 10 miles, then run 3 miles. The plan is to send the contestants out in a staggered fashion, via the following rule: the contestants must use the pool one at a time. In other words, first one contestant swims the 20 laps, gets out, and starts biking. As soon as this first person is out of the pool, a second contestant begins swimming the 20 laps; as soon as he or she is out and starts biking, a third contestant begins swimming, and so on.
Each contestant has a projected *swimming time*, a projected *biking time*, and a projected *running time*. Your friend wants to decide on a *schedule* for the triathlon: an order in which to sequence the starts of the contestants. Let's say that the *completion time* of a schedule is the earliest time at which all contestants will be finished with all three legs of the triathlon, assuming the time projections are accurate.
What is the best order for sending people out, if one wants the whole competition to be over as soon as possible? More precisely, give an efficient algorithm that produces a schedule whose completion time is as small as possible. Prove that your algorithm achieves this.

Solution:
Sort the athletes by decreasing biking + running time.

Proof:
The proof is similar to the proof we did for the scheduling problem to minimize maximum lateness. We first define an inversion as an athlete i with higher (bi+ri) being scheduled after athlete j with lower (bj+rj). We can then show that inversions can be removed without increasing the competition time. We then show that given an optimal solution with inversions, we can remove inversions one by one without affecting the optimality of the solution until the solution turns into our solution.

1- Inversions can be removed without increasing the competition time.
Remember that if there is an inversion between two items a and b, we can always find two adjacent items somewhere between a and b so that they have an inversion between them. Now we focus on two adjacent athletes (scheduled one after the other) who have an inversion between them, e.g. athlete i with higher (bi+ri) is scheduled after athlete j with lower (bj+rj). Now we show that scheduling athlete i before athlete j is not going push out the completion time of the two athletes i and j. We do this one athlete at a time:
    - By moving athlete i to the left (starting earlier) we cannot increase the completion time of athlete i
    - By moving athlete j to the right (starting after athlete i) we will push out the completion time of athlete j but since the swimming portion is sequential and athlete j gets out of the pool at the same time that athlete i was getting out of the pool before removing the inversion, and since athlete j is faster than athlete i in the biking and running sections, then the completion time for athlete j will not be worse than the completion time for athlete i prior to removing the inversion.

2- Since we know that removing inversions will not affect the completion time negatively, if we are given an optimal solution that has any inversions in it, we can remove these inversions one by one without affecting the optimality of the solution. When there are no more inversions, this solution will be the same as ours, i.e. athletes sorted in descending order of biking+running time. So our solution is also optimal.