Homework 6: Q3

Name: Xinkai Lin, xinkaili

*Don't forget to input your list of collaborators and sources on **AutoLab**.*
Please submit this file as a PDF.

1 Algorithm Idea

In this algorithm, giving input of n size array which contains the swim time, bike time and run time for ith camper. From the walk-through video, it's not hard to see that we want the slowest biking time + running time to go first due to the overlapping, then follow by the second slowest and so on. Then, I will sort the list by the slowest biking time + running time to the fastest. Now that we have the list sorted, we can output the best schedule based on the sorted list.

2 Algorithm Details

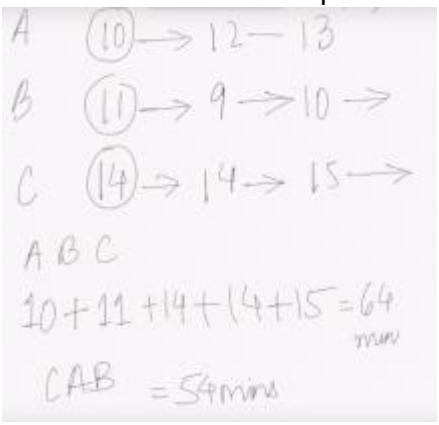
Input: n size array which contains the swim time, bike time and run time
For (i=0; i<n; i++) do
    for all camper compute time, $b_i + r_i$
sort the list according to the time of $b_i + r_i$ in a descending order
add each camper to the schedule

3 Proof of Correctness Idea

First of all, from the question giving and during the video we can see that there is only one person at a time in the pool, so the swimming time will be fixed, which will always be spending that much amount of time. Then, we will look at the biking and running time. Since biking and running can happen at the same time, so we want the slowest person to start first. When the new camper finished swimming the first person(slowest) have already start in advance and not wasting time. This way we can maximally using the time, and to come out with the most efficient schedule. Suppose we let the person who has fastest biking and running to start first, then by the time when he finished all the events, it didn't save much time because the fast people are finished to fast and left slowest person is still working just like the first example showing in the video. So, my algorithm is correct.

4 Proof Details

I will use the same example from the walk-through video.



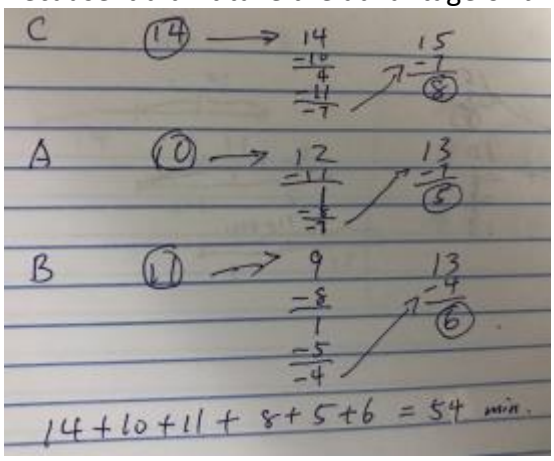
From this example, we can see that for ABC the bike time + run time as follow:

A = 12 + 13 = 25

B = 9 + 10 = 19

C = 14 + 15 = 29

As the result, we can see that the first example ABC it is way longer than the CAB. Because it didn't take the advantage of bike and run can happen simultaneously.



By using this algorithm, we are preforming CAB and we only spent 54 min. I have shown in the drawing above, the number with cycle is the time what we actually spending. Since the bike and run can happen simultaneously, so we want the slowest person to be starting as soon as possible, so they can start in an advance which going to save more times.

5 Runtime Analysis

The runtime for this algorithm is going to be O(nlog(n)). At the beginning, I use the for loop which to find out the total time of bike + run of each person, this will take O(n) time, since it will loop through n people and each will perform O(1) time. Then, I use the sort function base on the time that I found, the sort function will run at O(nlog(n)). So, the final runtime for this algorithm is O(n) + O(nlog(n)) which is O(nlog(n)).