CS41 Homework 5

This homework is due at 11:59PM on Wednesday, October 19. Write your solution using LaTeX. Submit this homework in a file named `hw5.tex` using **github**.

This is a **one week, 8 point assigment**. You are not expected to work on this assignment over fall break. This is a partnered homework. You should primarily be discussing problems with your homework partner.

It's ok to discuss approaches at a high level with others. However, you should not reveal specific details of a solution, nor should you show your written solution to anyone else. The only exception to this rule is work you've done with a lab teammate *while in lab*. In this case, note (in your **homework submission poll**) who you've worked with and what parts were solved during lab.

The main **learning goals** of this homework assignment are to algorithm design and analysis for greedy algorithms.

1. **Summer camp triathlon** (Kleinberg and Tardos, 4.6) Your friend is working as a camp counselor, and is in charge of organizing activities for a set of junior-high-school-age campers. One of the 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 person is out of the pool, a second contestant begins swimming the 20 laps; as soon as the second person is out and starts biking, a third contestant begins swimming, and so on.)

   Each contestant has a projected *swimming time* (the expected time it will take them to complete the 20 laps), a projected *biking time* (the expected time it will take them to complete the 10 miles of bicycling), and a projected running time (the expected time it will take them to complete the 3 miles of running). Your friend wants to decide on a *schedule* for the triathalon: 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 triathalon, assuming they each spend exactly their projected swimming, biking, and running times on the three parts. (Again, note that participants can bike and run simultaneously, but at most one person can be in the pool at any time.) What's the best order for sending people out, if one wants the whole competition to be over as early as possible? More precisely, give an efficient algorithm that produces a schedule whose completion time is as small as possible.

2. **Scheduling Video Streams.** Suppose you have n video streams that need to be sent, one after another, over a communication link. Stream i consists of a total of b_i bits that need to be sent, at a constant rate, over a period of t_i seconds. You cannot send two streams at the same time, so you need to determine a *schedule* for the streams: an order in which to send them. Whichever order you choose, there cannot be any delays between the end of one stream and the start of the next. Suppose your schedule starts at time 0 (and therefore ends at time $\sum_{i=1}^{n} t_i$, whichever order you choose). Assume that all the values b_i and t_i are positive integers.

   Because you're just one user, the owner of the communication link does not want you taking up too much bandwidth, so it imposes the following constraint, using a fixed parameter r:

Constraint. *For each natural number $t > 0$, the total number of bits you send over the time interval from 0 to t cannot exceed rt.*

We say that a schedule is *valid* if it satisfies the constraint imposed by the owner of the communication link.

In this problem, your goal is to design an algorithm that, given an input of n streams, each specified by its number of bits b_i and its time duration t_i, as well as the link parameter r, outputs YES if there exists a valid stream schedule and NO if there is no valid stream schedule.

(a) Consider the following claim:

Claim 1. *There exists a valid schedule if and only if each stream i satisfies $b_i \leq rt_i$.*

Decide whether you think the claim is TRUE or FALSE. If the claim is TRUE, give a proof the claim is TRUE. If you think the claim is false, prove the claim is FALSE.

(b) Give an algortihm `StreamScheduler(streams, r)` that takes a List of n streams, each specified by $[b_i, t_i]$ its total number of bits b_i and its time duration t_i, as well as the link parameter r, and determines whether there exists a valid schedule. Your algorithm should output YES if there exists a valid schedule, and NO otherwise. The runtime of your algorithm should be polynomial in n.