**Code Jam 2017 - Round 1B**

# Analysis: Steed 2: Cruise Control

Pop quiz, hotshots! This problem seems pretty complicated at first glance. What do you do?

One natural strategy is to try binary searching on Annie's speed, but it is difficult to directly determine whether a given speed avoids passing another horse; the input data alone does not tell us where each horse is at any given time, because horses might slow other horses down. In theory, we could figure out when faster horses catch up to slower horses and slow down, determine the exact path of each horse, and check whether our chosen speed crosses any of those paths. With only up to two horses in test set 1, this sort of calculation is feasible, but it would be laborious for test set 2.

However, we can avoid all of that work via some observations. To maximize cruising speed, Annie's horse should reach the destination at exactly the same time as the horse ahead of her (let's call it Horse A); there is no reason to leave a gap. Either Horse A will reach the destination without having to slow down (and so it will be the one that directly limits Annie's speed), or it will be slowed down at some point by the horse ahead of it (let's call it Horse B). The same is true for Horse B: either it will never have to slow down (and so it will be the one that ultimately limits Annie's speed), or it will be slowed down by the horse ahead of *it*, and so on. So there will be a single "limiting horse" on the road that ultimately determines how fast Annie's horse can reach the destination. We claim that this "limiting horse" is the only horse that matters, and we can disregard all of the others!

It is easy to see that we can ignore the horses to the east of the limiting horse; they will reach and pass the destination before the limiting horse gets there. What about the "intermediate horses" between Annie and the limiting horse? We know from the way we have defined the limiting horse that every intermediate horse will catch up to the limiting horse before reaching the destination. (If one did not, then *it* would be the limiting horse.) Suppose that Annie chooses a cruising speed that gets her to the destination at exactly the same time as the limiting horse. We certainly cannot go faster than this. Moreover, this speed is safe: it cannot possibly cause Annie to pass any of the intermediate horses. If she were going fast enough to overtake an intermediate horse, then she would definitely be going fast enough to pass the limiting horse, since every intermediate horse will catch up to the limiting horse. This would cause a contradiction. Therefore, we do not need to worry about the intermediate horses or their interactions with each other.

So, once we have identified the limiting horse, the strategy is simple: go at the exact speed that will cause Annie to reach the destination at the same time as the limiting horse. This speed can be found in constant time. We could identify the limiting horse directly via the argument in our third paragraph above, but even this would be unnecessary work. Instead, for each horse in turn, we can pretend that it is the limiting horse and calculate the cruising speed that it would force. Then the smallest of those speeds is our answer. (If any horse allows a faster cruising speed than another, it cannot be the limiting horse, because that cruising speed would cause Annie to pass the true limiting horse.) This takes O(**N**) time.