Anthony Torres

CS 323

Fall 2019 Assignment Writeup

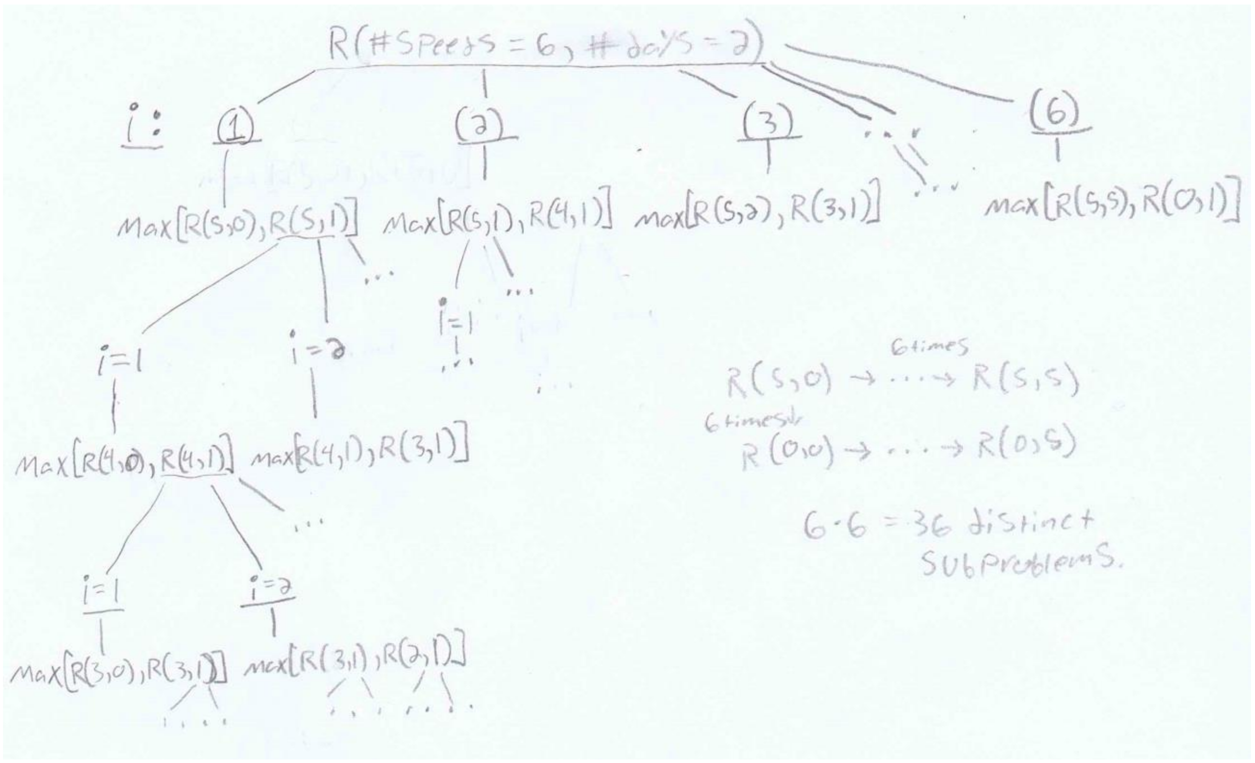
1. Running Trials and Tribulations

(a) Describe the optimal substructure/recurrence that would lead to a recursive solution

At the base case state, the only two cases for the runner are whether they get injured on a day or not. If the runner is injured at a speed, you only need to check the previous speeds with one less day. If the runner is not injured, you would have to check the remaining speeds with the same amount of days.

(b) Code your recursive solution under runTrialsRecur(int possibleSpeeds, int days). If your recursive function runs forever, in order for grading to happen quickly please comment out the code progress you made instead.

(c) Draw recurrence tree for given (# speeds = 6, # days = 2)



(d) How many distinct subproblems do you end up with given 6 speeds and 2 days?

There would be (6*6) = 36 distinct subproblems.

(e) How many distinct subproblems for N speeds and M days?

There would be (n*n) = n^2 distinct subproblems.

(f) Describe how you would memoize runTrialsRecur.

Create a table to store the solutions for each distinct subproblem. When calling the runTrialsRecur function, the table will be searched to see if either the left or right recursive call is stored in the table already. If it is stored then you would only need to look up the table at a constant speed, instead of solving the subproblem again.

(g) Code a dynamic programming bottom-up solution runTrialsBottomUp(int possibleSpeeds, int days)

(h) Extra Credit: 15 pts, write a function that will also print which speeds the athlete should test during those minimum number of speedtest runs, and have it print to the output