Homework 4: Recursive Algorithms

Due: Friday, February 21 at 5pm on Canvas

Concepts: recursion, divide-and-conquer, generating random instances

Theory

1. (9 points) Olin's dining hall is considering implementing a new dining format to improve efficiency. Instead of having stations, all the food options will be available in a single line. Further, students must choose a continuous interval of foods to take. That is, if the available options are [french fries, brussel sprouts, chicken sandwiches, tomato soup, fruit salad], then a student could take french fries, brussel sprouts, and a chicken sandwich, but not just the fries and sandwich.

   Luckily, you can use your DSA skills to find the optimal food interval for you. Suppose that there are n foods available, and let h_i be the happiness value for item i (possibly negative if you dislike a food). Give a divide-and-conquer or recursive algorithm to find the interval $[i, i+1, \ldots, j-1, j]$ that maximizes the sum of happiness scores $\sum_{k=i}^{j} h_i$. In a few sentences, convincingly argue why your algorithm finds an optimal solution, and analyze the runtime of your algorithm (it might help to write pseudocode to do the latter.) The description of your algorithm should be clear and precise enough that I could write code for it just given your solution.

2. (6 points) The students at Olin are suffering a new version of the freshman plaque called academitis. Fortunately, this unique bug can only be passed between students during class time and there's a known cure – leweekend. Suppose that you want to distribute the cure to all potentially affected students and you've identified patient zero that started the outbreak. Design an your algorithm to find this list. Be sure to give a clear description and mention any supporting data structures you use. Then, in a few sentences, argue the correctness of the output. You do *not* have to analyze the runtime, and you may assume that you can find a list of a student's classmates.

   Hint: Look back at the maze example from class.

Practice

Analyzing the output of an algorithm on different instances can yield insight into the behavior of the algorithm. For this assignment, you will generate random inputs using the `numpy.random` package in python and compare the results. Documentation: `https://docs.scipy.org/doc/numpy-1.14.0/reference/routines.random.html`.

3. (3 points) Implement your algorithm from problem 1 and a corresponding test function.

4. (3 points) Generate 100 random instances of happiness value lists of length $n = 100$ where each value h_i is drawn uniformly between -10 and 10. This represents a situation in which you have a wide and evenly distributed set of preferences. Record the average length and value of the max interval returned by your function. Comment on the results.

5. (3 points) Now, repeat the same experiments but this time each value h_i with probability 0.7 is a normal random variable with mean 6 and standard deviation 1 and with probability 0.3 is a normal random variable with mean -7 and standard deviation 0.5. In this case, you are sometimes very picky but often pretty happy. Again, comment on the results and compare to your results from above.