Homework 2

OPTIONAL PROBLEMS
(No due date)

1 Written Problems

1.1 Longest Increasing Subset

NOTE: this is a challenge problem
Given an array of integers, find the length of the longest increasing subsequence of these integers. (This can be done with dynamic programming in O(nlogn) time!!) For example,

1. \([0, 3, 6, 2, 10, 1, 5, 33]\) would return 5, because the longest increasing subsequence, \([0, 3, 6, 10, 33]\) is of length 5
2. \([14, 2, 15, 11]\) would return 2, because the longest increasing subsequences, \([14, 15]\), \([2, 11]\) and \([2, 15]\) are all of length 2

2 Python Problems

2.1 Climbing Stairs

Given a set of stairs comprised of \(n\) steps, and that you can climb 1, 2, or 3 steps at a time, write a function that returns the number of ways to climb that set of stairs. For example,

1. climber(2) would return 2, because the sequence of steps could be \([1, 1]\) or \([2]\)
2. climber(3) would return 4, because the sequence of steps could be \([1, 2]\), \([2, 1]\), \([3]\), or \([1, 1, 1]\)

You do not need to return the possible sequences of steps, just the number.

2.2 Climbing Stairs++

Modify your climber(n) function to print out the possible sequences of steps, such that climber(3) would print out “\([1,2]\), \([2,1]\), \([3]\), \([1,1,1]\)”