11/7/16

FOCS Day 19

3. a) • max_subarray([-2,1,-3,4,-1,2,1,-5,4])

And then it doesn't really do anything else. It's an iterative solution, not a recursive one

b) No. There is no multi-branch recursion, because recursion isn't involved

c) Yes- the iteration solves a smaller problem the 1st time, then adds a number, and checks whether the solution has changed. It could just compute the sum of every subarray

d) It isn't recursive. No recursion takes place. This is a loop. I am confused.

4. a)

• bs(8, [1,3,4,6,7,8,10,13,14])

↓

• bs(3, [1,3,4,6,7])

↓

• bs(3, [1,3])

↓

• bs(3, [3])

b) It is not - it doesn't branch, just cleverly cuts down on possibilities (the possibility space of runs it could do branches, but the program itself does not)

c) No- it doesn't employ memoization, because each sub-problem only needs to be solved once

d. ~~I'm not entirely sure what to memorize. There are no repeated subproblems~~ It's a straight line again, the memoization is only useful on later runs

e. It just recurses in a straight line. I don't know how to improve on that

---

1. a) 1. No: there is no letter that loves all other letters
2. Yes; C is loved by every letter
3. Yes: all letters love at least one letter
4. Yes: every letter is loved by at least one letter
5. Yes: the table has at least 1 entry
6. No: Inverse of 5, really. All letters love something.
7. Yes: there is an x that doesn't love every y

b)

	Table 1	$X \geq y$	$X = y$
1.	False	True	True
2.	False	False	True
3.	False	True	True