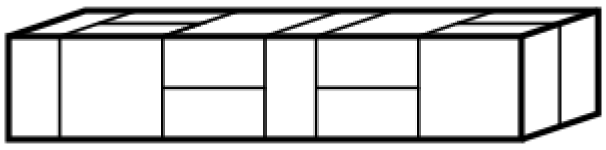
Started on	Wednesday, 23 September 2020, 2:50 PM
State	Finished
Completed on	Wednesday, 23 September 2020, 3:10 PM
Time taken	19 mins 49 secs
Grade	Not yet graded

Question **1**
Complete
Marked out of 4.00

[SUBJECTIVE] A slab is a three dimensional box with dimensions 1 x 2 x 2, 2 x 1 x 2, or 2 x 2 x 1. We want to compute the number of different ways to fill a 2 x 2 x n box with n slabs.



a $2 \times 2 \times 10$ box filled with ten slabs.

Derive a recurrence relation for the solution and write a $O(n)$ time program to compute this solution.

Volume of the box = 4n units

volume of the slab is 4 units.

So n slabs will fill the box.

left half can be filled with n/2 slabs and right half with n/2 slabs.

So it seems to be that the recurrence relation would look like

T(n) = 2 * T(n/2) + O(1)

a = 2, b^2 = 2^0 = 1

so T👎 = O(n^logb(a)) = O(n^1)

The combine step takes O(1) because we just have to multiply the number of ways of filling left half by number of ways of filling right half.

```
number_of_ways👎
      volume_of_box = 4*n
      num_ways = 0
       if volume_of_box == 4:
             return 3
       else num_ways += 2 * number_of_ways(volume_of_box/2)
```

Question **2**
Complete
Marked out of 2.00

[Subjective] Let us call a sequence of integers B[1..k] *bumpy* if B[i] < B[i+1] for all even i and B[i] > B[i+1] for all odd i. We want to find the length of the longest bumpy subsequence of a sequence A[1..n] of n integers. Give a simple *recursive* definition for the function *lbs(A[1..n]),* which computes this value.

Write only the final recursive definition.

lbs(A[1...n]):

Question **3**

Complete

Marked out of 3.00

[Subjective] For this problem we will consider a variant of the Stable Matching problem where men and women can be *indifferent* between certain choices of partners. Just as before, there are n men and n women, but there can be *ties* in the preference lists. For e.g. with n=4, a woman could say that m_1 is ranked in the first placed; second place is a tie between m_2 and m_3 (she has no preference between them); and m_4 is in the last place. We will say that w prefers m to m' if m is ranked higher than m' on her preference list (they are not tied).

In such a scenario, a *weak instability* in a perfect matching S consists of a man m and a woman w such that their partners in S are w' and m', respectively, and one of the following holds:

- m preferes w to w', and w either prefers m to m' or is indidfferent between these two choices; or
- w prefers m to m', and m either prefers w to w' or is indifferent between these two choices.

In other words, the pairing between m and w is either preffered by both, or preferred by one while the other is indifferent.

Does there always exist a perfect matching with no weak instability? Either give an example of a set of men and women with preference lists for which every perfect matching has a weak instability; or give an algorithm that is guaranteed to find a perfect matching with no weak instability.

when m proposes if w is already engaed to m' and she is indifferent to m and m' , she should accept m's proposal.
This I think would give a stable matching with no weak instability

gale_shapeley(men, women):

freemen = {men}

man = a freeman in freemen

while man is yet to propose to a woman and is free

man chooses the woman highest in his pref list and proposes to her

if w is free:

(m,w) get engaged

if w is already engaged to m':

if w preferes m over m' **or** if w is indifferent to m and m':

w breaks engagement with m' and gets engaged to m

m' becomes free

else:

w rejects m

m proposes to next woman in his list.

◄ Mid Semester Examination: Objective

Jump to...