081203M04001H - Algorithm Design and Analysis

Assignment 2

October 15, 2021

Notice:

1. Please submit your answer in hard copy AND submit a digital version to UCAS website http://sep.ucas.ac.cn.
2. Hard copy should be submitted before 9 am. October 29 and digital version should be submitted before 11:30 pm. October 29.
3. You can choose **three** from problems 1-5, and you should do at least the following things:
   (a) Describe the optimal substructure and DP equation;
   (b) Describe your algorithm in daily language or pseudo-code;
   (c) Prove the correctness of your algorithm;
   (d) Analyse the complexity of your algorithm.

1 Money robbing

A robber is planning to rob houses along a street. Each house has a certain amount of money stashed, the only constraint stopping you from robbing each of them is that adjacent houses have security system connected and it will automatically contact the police if two adjacent houses were broken into on the same night.

1. Given a list of non-negative integers representing the amount of money of each house, determine the maximum amount of money you can rob tonight without alerting the police.
2. What if all houses are arranged in a circle?

2 Largest Divisible Subset

Given a set of distinct positive integers, find the largest subset such that every pair (S_i, S_j) of elements in this subset satisfies: $S_i\%S_j = 0$ or $S_j\%S_i = 0$.

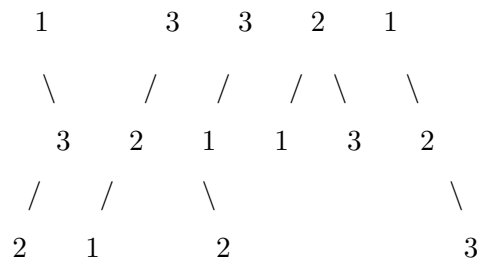
Please return the largest size of the subset.

Note: $S_i\%S_j = 0$ means that S_i is divisible by S_j.

3 Unique Binary Search Trees

Given n, how many structurally unique BST's (binary search trees) that store values 1...n?

Explanation: Given n = 3, there are a total of 5 unique BST's:



4 Word Break

Given a string S and a dictionary of words, determine if the string S can be segmented into a space-separated sequence of one or more dictionary words.

Note: Each word in the dictionary may be reused multiple times in the segmentation. You can return TRUE if the string S is empty.

5 Distinct Sequences

Given two strings S and T, return the number of distinct subsequences of S which equals T.

A string's subsequence is a new string formed from the original string by deleting some (can be none) of the characters without disturbing the remaining characters' relative positions. (i.e., "ACE" is a subsequence of "ABCDE" while "AEC" is not).