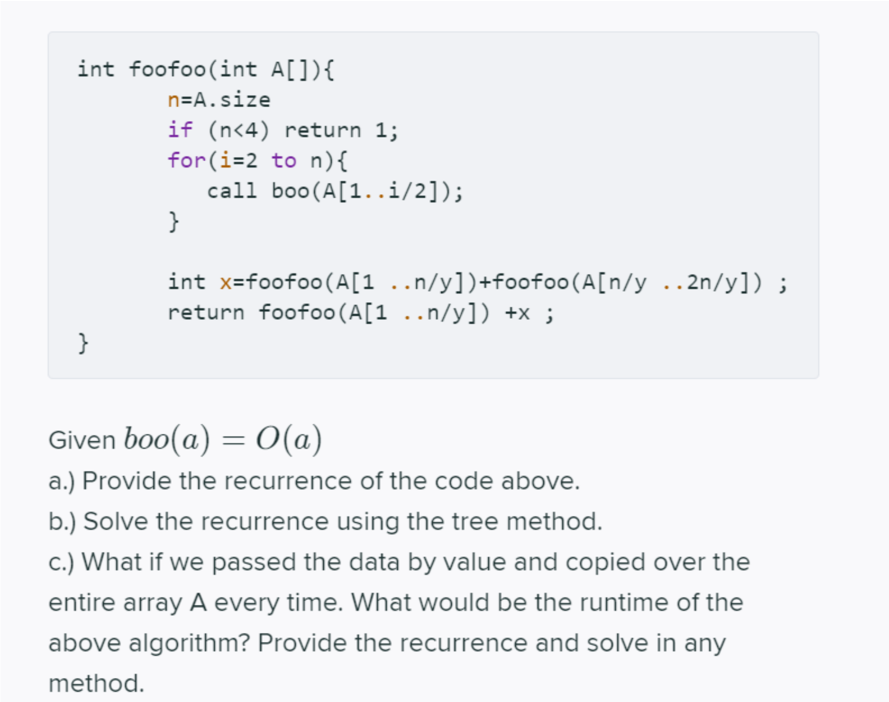
Sample Questions from past tests

Some Divide and Conquer questions



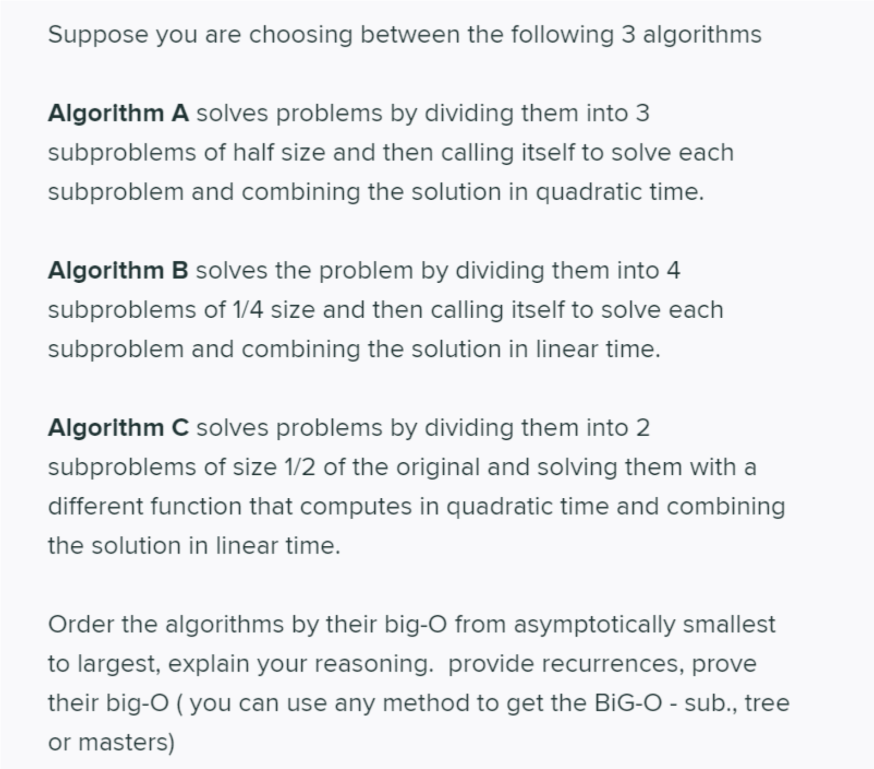
```
int foofoo(int A[]){
       n=A.size
       if (n<4) return 1;
       for(i=2 to n){
          call boo(A[1..i/2]);
       }

       int x=foofoo(A[1 ..n/y])+foofoo(A[n/y ..2n/y]) ;
       return foofoo(A[1 ..n/y]) +x ;
}
```

Given $boo(a) = O(a)$

a.) Provide the recurrence of the code above.

b.) Solve the recurrence using the tree method.

c.) What if we passed the data by value and copied over the entire array A every time. What would be the runtime of the above algorithm? Provide the recurrence and solve in any method.



Suppose you are choosing between the following 3 algorithms

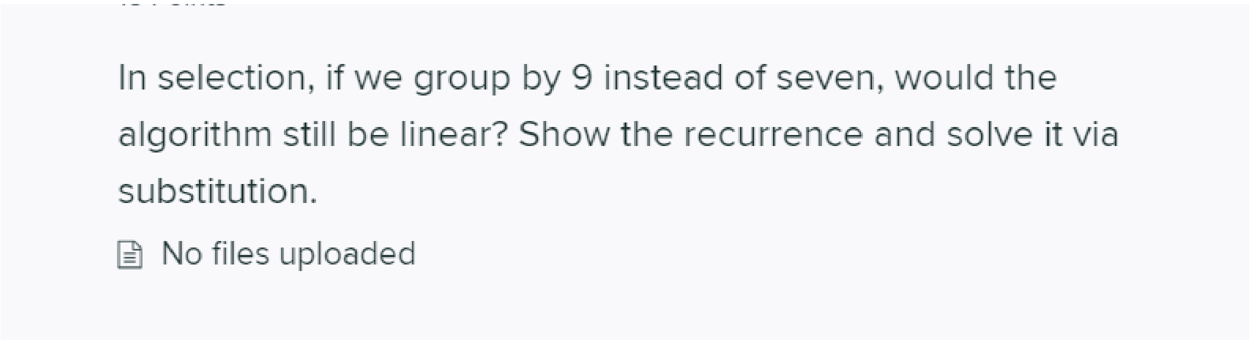
Algorithm A solves problems by dividing them into 3 subproblems of half size and then calling itself to solve each subproblem and combining the solution in quadratic time.

Algorithm B solves the problem by dividing them into 4 subproblems of 1/4 size and then calling itself to solve each subproblem and combining the solution in linear time.

Algorithm C solves problems by dividing them into 2 subproblems of size 1/2 of the original and solving them with a different function that computes in quadratic time and combining the solution in linear time.

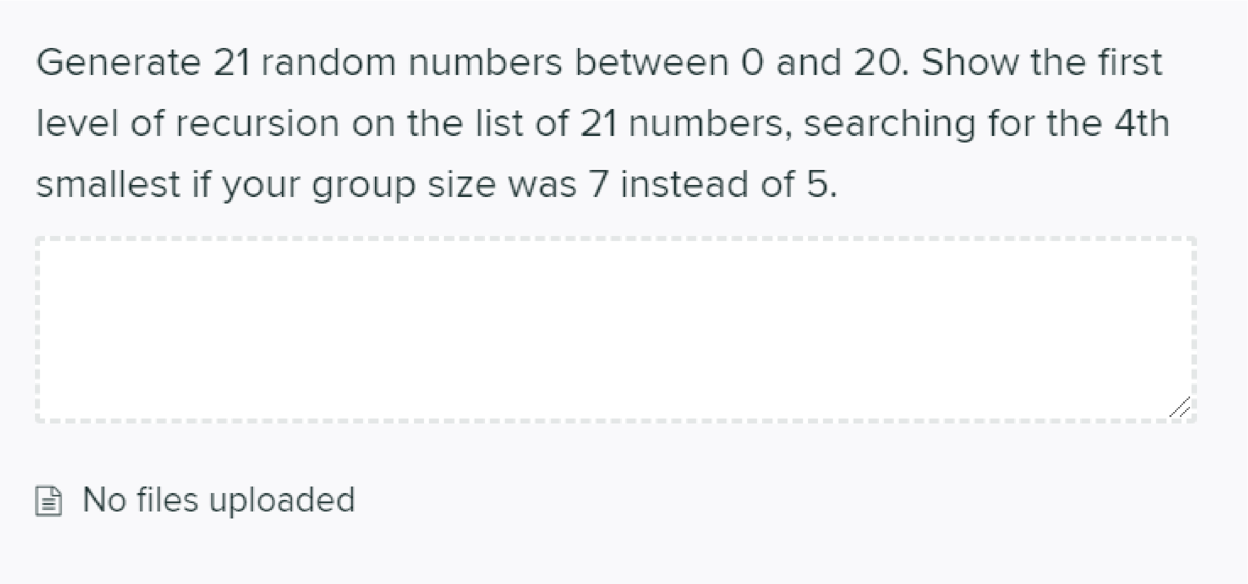
Order the algorithms by their big-O from asymptotically smallest to largest, explain your reasoning. provide recurrences, prove their big-O (you can use any method to get the BiG-O - sub., tree or masters)

Selection questions



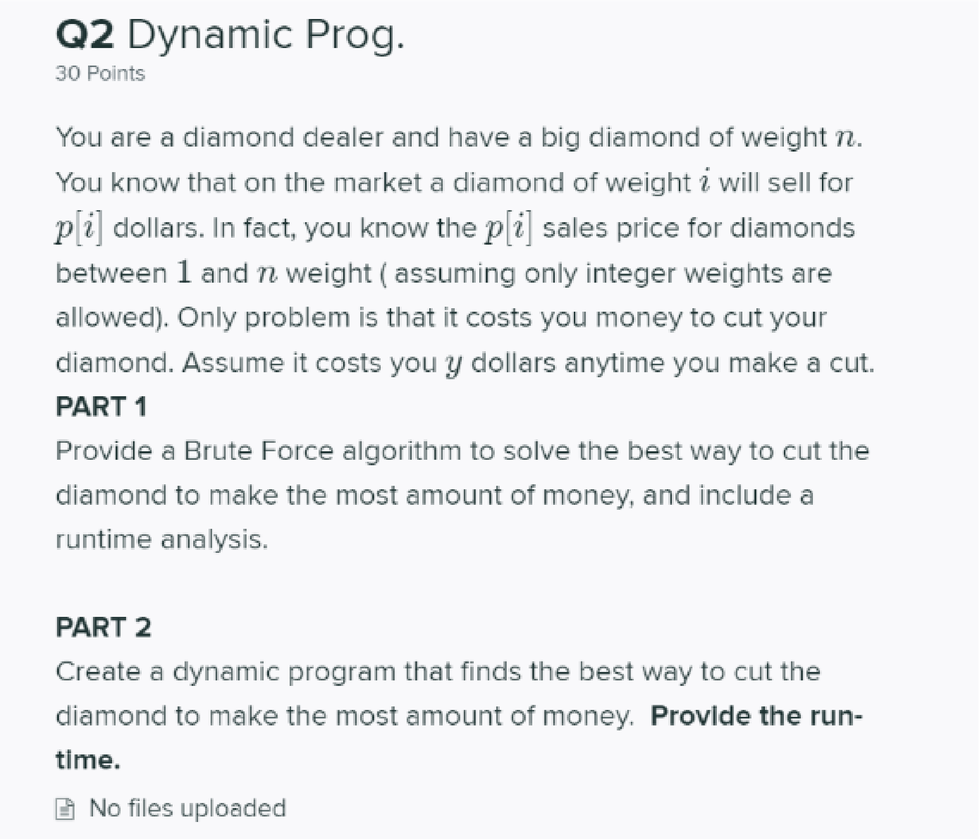
In selection, if we group by 9 instead of seven, would the algorithm still be linear? Show the recurrence and solve it via substitution.

No files uploaded



Generate 21 random numbers between 0 and 20. Show the first level of recursion on the list of 21 numbers, searching for the 4th smallest if your group size was 7 instead of 5.

Dynamic programming



Q2 Dynamic Prog.

30 Points

You are a diamond dealer and have a big diamond of weight n. You know that on the market a diamond of weight i will sell for $p[i]$ dollars. In fact, you know the $p[i]$ sales price for diamonds between 1 and n weight (assuming only integer weights are allowed). Only problem is that it costs you money to cut your diamond. Assume it costs you y dollars anytime you make a cut.

PART 1

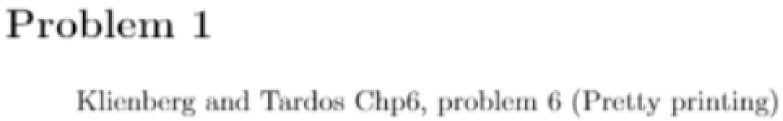
Provide a Brute Force algorithm to solve the best way to cut the diamond to make the most amount of money, and include a runtime analysis.

PART 2

Create a dynamic program that finds the best way to cut the diamond to make the most amount of money. **Provide the run-time.**

No files uploaded

Include both the recurrence and the pseudocode.



Problem 1

Klienberg and Tardos Chp6, problem 6 (Pretty printing)

Problem 3

Shuffle: Suppose you are given three strings of characters: $X = x_1x_2...x_n$, $Y = y_1y_2...y_m$, and $Z = z_1...z_{n+m}$. We want to know whether we can shuffle X and Y to produce Z. A shuffle S of two strings X and Y is valid if the characters of X appear in order in S and if we remove those characters from S, what remains is Y . For example, if $X = aabacb$, $Y = caa$, then S *caaabaacb* is a shuffle (the first, second and 7th characters give us Y , the rest is X), but *ccaaaabab* is not (since we can't have two c's appear before any b's).

a Give an brute force algorithm that given X, Y, Z determines whether Z is a shuffle of X and Y. Analyze your algorithms space and time requirements.

b Give an efficient dynamic programming algorithm that given X, Y, Z determines whether Z is a shuffle of X and Y . Analyze your algorithms space and time requirements.