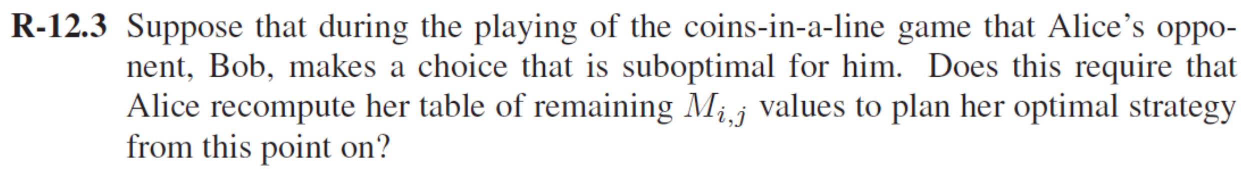
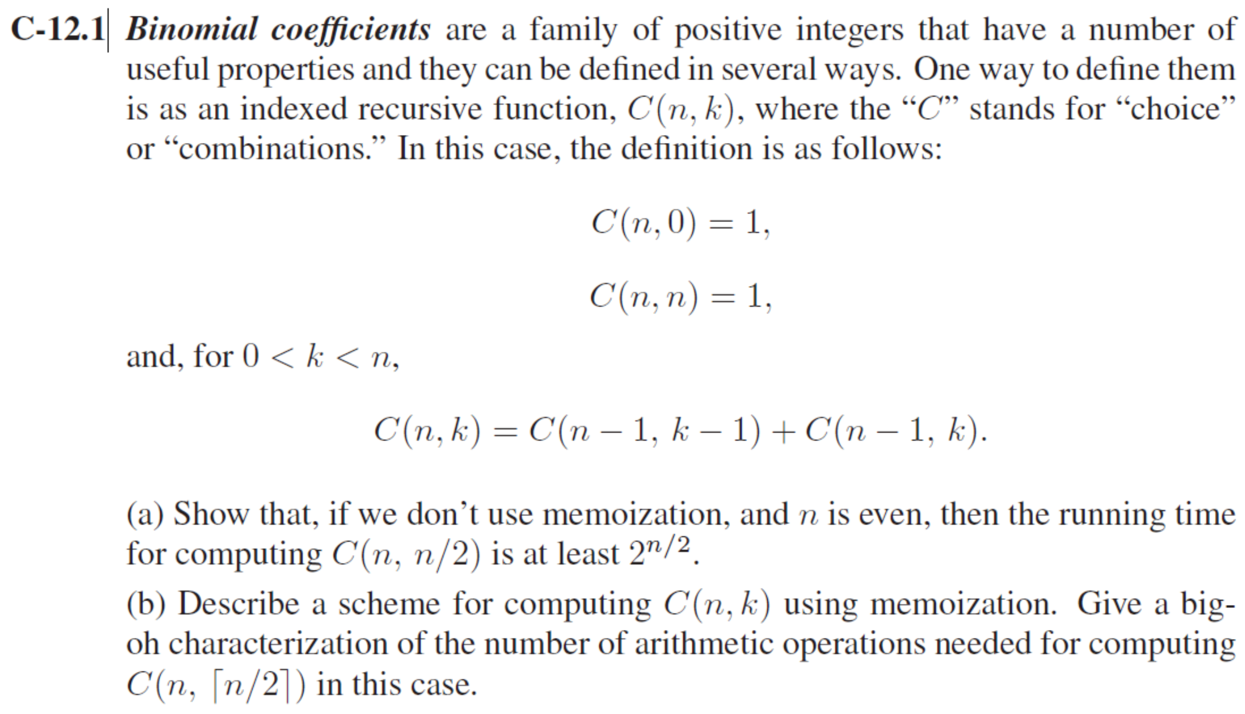
3120homework04

Class	CPSC 3120
Due Date	@April 23, 2025
Status	Done
Created time	@February 4, 2025 12:45 AM



R-12.3 Suppose that during the playing of the coins-in-a-line game that Alice's opponent, Bob, makes a choice that is suboptimal for him. Does this require that Alice recompute her table of remaining $M_{i,j}$ values to plan her optimal strategy from this point on?

No, Alice does not need to recompute her table of remaining Mi,j values if Bob makes a suboptimal move. The dynamic programming table Mi,j is built based on the assumption that both players play optimally from every position. Even if Bob makes a mistake, Alice can still rely on the existing table to choose her next move, because it tells her the best outcome she can achieve from any subarray, regardless of how they got to that state. So Alice can treat the new game state as a smaller subproblem already solved in the table and continue playing optimally from there.



C-12.1 ***Binomial coefficients*** are a family of positive integers that have a number of useful properties and they can be defined in several ways. One way to define them is as an indexed recursive function, $C(n, k)$, where the "C" stands for "choice" or "combinations." In this case, the definition is as follows:

$$C(n, 0) = 1,$$

$$C(n, n) = 1,$$

and, for $0 < k < n$,

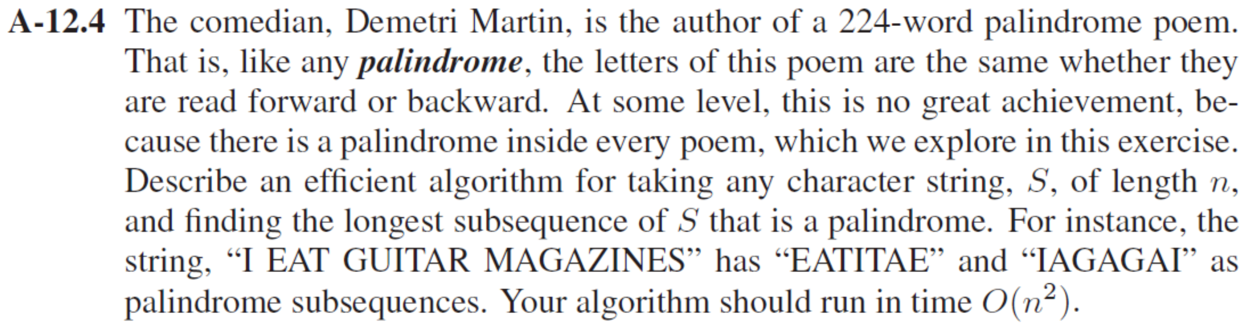
$$C(n, k) = C(n-1,\ k-1) + C(n-1,\ k).$$

(a) Show that, if we don't use memoization, and n is even, then the running time for computing $C(n,\ n/2)$ is at least $2^{n/2}$.

(b) Describe a scheme for computing $C(n, k)$ using memoization. Give a big-oh characterization of the number of arithmetic operations needed for computing $C(n,\ \lceil n/2 \rceil)$ in this case.

(A) if we compute C(n,n/2) without memoization, the recursive calls branch exponentially. At each level of recursion, for 0<k<n we compute C(n,k)=C(n-1,k-1)+C(n-1,k). This creates a binary recursion tree. For C(n,n/2) the height of the tree is nearly n/2, and each level nearly doubles the number of calls. So the number of calls grows at least as fast as 2^(n/2) meaning the runtime is at least omega(2^(n/2)).

(B) To compute C(n,k) efficiently, we can store the results of problems in a hashmap so each problem is only computed once. With memoization, there are O(nk) problems, and each is computed once. So for C(n,ceil(n/2)), the number of arithmetic operations is O(n^2) since k≤n and every pair (n,k) is computed maximum once.



A-12.4 The comedian, Demetri Martin, is the author of a 224-word palindrome poem. That is, like any ***palindrome***, the letters of this poem are the same whether they are read forward or backward. At some level, this is no great achievement, because there is a palindrome inside every poem, which we explore in this exercise. Describe an efficient algorithm for taking any character string, S, of length n, and finding the longest subsequence of S that is a palindrome. For instance, the string, "I EAT GUITAR MAGAZINES" has "EATITAE" and "IAGAGAI" as palindrome subsequences. Your algorithm should run in time $O(n^2)$.

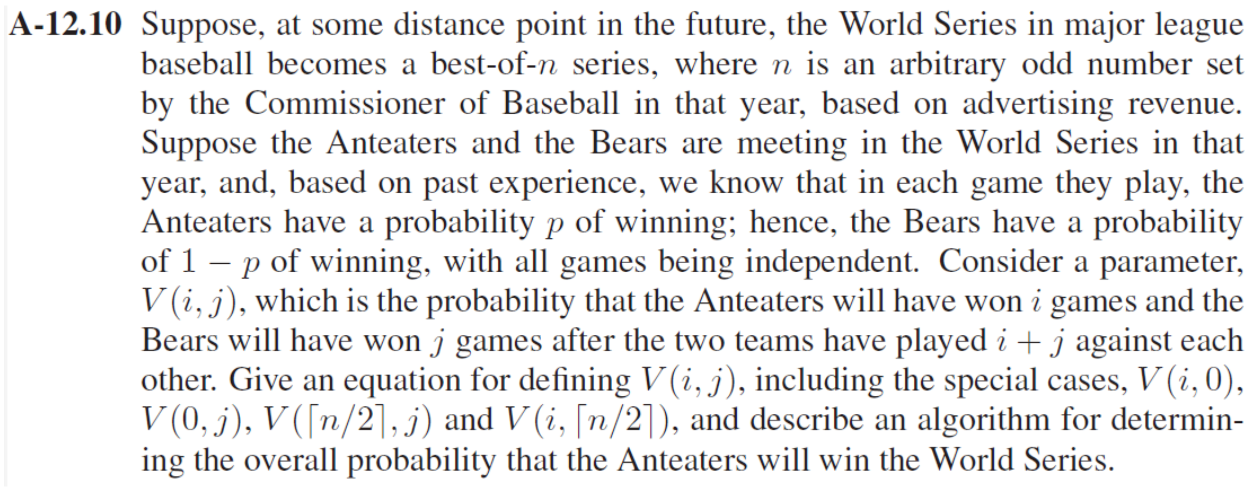
let p[i][j] be the length of the longest palindrome subsequence in the substring S[i,j].

If i==j, a single character is a palindrome → p[i][i]=1.

If S[i]==S[j], the characters can be matched→p[i][j]=p[i+1][j-1]+2

Else skip S[i] or S[j]→p[i][j]=max(p[i+1][j],p[i][j-1])

This fills an nxn table where each cell takes O(1) time, making an O(n^2) algorithm.



A-12.10 Suppose, at some distance point in the future, the World Series in major league baseball becomes a best-of-n series, where n is an arbitrary odd number set by the Commissioner of Baseball in that year, based on advertising revenue. Suppose the Anteaters and the Bears are meeting in the World Series in that year, and, based on past experience, we know that in each game they play, the Anteaters have a probability p of winning; hence, the Bears have a probability of $1 - p$ of winning, with all games being independent. Consider a parameter, $V(i, j)$, which is the probability that the Anteaters will have won i games and the Bears will have won j games after the two teams have played $i + j$ against each other. Give an equation for defining $V(i, j)$, including the special cases, $V(i, 0)$, $V(0, j)$, $V(\lceil n/2 \rceil, j)$ and $V(i, \lceil n/2 \rceil)$, and describe an algorithm for determining the overall probability that the Anteaters will win the World Series.

Let w = ceil(n/2), the number of wins needed to win the series

Base cases:

if i == w, Anteaters already won, V(i,j)=1

if j == w, Bears already won, V(i,j)=0

Recursive case:

V(i,j)=p*V(i+1,j)+(1-p)*V(i,j+1)

If anteaters win the next game, transition to (i+1,j)

If bears win the next game, transition to (i,j+1)

computing V(0,0) will give us the probability that the anteaters will win the series starting from 0-0