

Atividade Avaliativa 4 – Programação Dinâmica (parte 1)

1. Explique o que é e como funciona a programação dinâmica.
2. Escreva uma função recursiva para calcular a o n-ésimo termo da série de Fibonacci. Calcule a complexidade dessa função e explique porque ela não é eficiente.
3. Explique como podemos utilizar a programação dinâmica para desenvolver algoritmos mais eficientes para a sequência de Fibonacci:
 - a) Abordagem Top-Down (memorização)
 - b) Abordagem Botton-Up (reversão)
4. Projete um algoritmo recursivo para o problema da sequência de cédulas, sem a utilização de programação dinâmica. Calcule a complexidade da sua função e explique se ela eficiente.
5. Ainda sobre o problema da sequência de cédulas, utilize programação dinâmica com uma abordagem Bottom-Up (não recursiva) para desenvolver uma função mais eficiente. Calcule a complexidade da nova função e compare com a função recursiva.
6. Considere a sequência de $n = 12$ cédulas a seguir: $C = [2, 5, 5, 2, 10, 50, 100, 50, 20, 20, 50, 100]$
 Sabendo que F é um vetor em que $F[0] = 0$ e $F[1] = c1$, execute manualmente o algoritmo desenvolvido no exercício 5 para obter a solução do problema, ou seja, o máximo valor de dinheiro que pode ser coletado sem que 2 cédulas vizinhas sejam obtidas. Você deve gerar todos os valores de $F[i]$ para I iniciando em 2 e terminando em n .
7. Projete um algoritmo para o problema do robô coletor de moedas usando programação dinâmica. Calcule a complexidade da sua função e explique se ela eficiente.
8. Considere o seguinte tabuleiro de entrada para o problema do robô coletor de moedas.

	1	2	3	4	5	6
1			●		●	
2	●	●		●		
3		●			●	
4			●			●
5		●	●		●	
6			●		●	

	1	2	3	4	5	6
1						
2						
3						
4						
5						
6						

Execute o algoritmo desenvolvido no exercício 7 para solucionar essa instância do problema do robô coletor de moedas. Calcule todos os valores de F , ou seja, preencha o quadro da direita. Qual é o trajeto que deve ser percorrido pelo robô iniciando na posição (1, 1)?

9. O problema do corte da haste nos diz que dada uma haste de comprimento n e uma tabela de preços para cada possível pedaço da haste, devemos maximizar o ganho. Projete um algoritmo recursivo para o problema do corte da haste usando programação dinâmica. Calcule a complexidade da sua função e explique se ela é eficiente.

10. Considerando o algoritmo projetado no exercício anterior, resolva a seguinte instância do problema do corte da haste:

l_i	1	2	3	4	5	6
p_i	2	6	9	10	12	16

Suponha uma haste metálica com $n=6$ possíveis pedaços. Mostre como o algoritmo obtém a solução ótima.

"Ser feliz não é ter o que você quer, mas querer o que você tem."
(Autor Anônimo)