

Atividade da Disciplina CC5661

Guilherme Wachs Lopes

1 Descrição

Sabe-se que muitos problemas na computação podem ser resolvidos utilizando técnicas de programação, tais como: Divisão e Conquista, Programação Dinâmica e Algoritmos Gulosos. Essas técnicas ajudam o desenvolvedor a modelar e criar soluções que são computacionalmente mais eficientes (tanto em termos de tempo quanto de memória) que a estratégia mais trivial (ou força-bruta).

Na Seção 3 você encontrará uma lista de problemas que podem ser resolvidos utilizando uma (ou mais) técnicas de programação. O presente trabalho de disciplina tem por objetivo discutir, modelar, implementar e fazer a análise assintótica de 3 problemas (escolhidos a partir da Seção 3).

A implementação dos algoritmos poderá ser feita em qualquer linguagem de programação

Para cada problema, o mesmo deve abordado em 4 diferentes seções:

- Explicação do problema: deve descrever o problema e dar exemplos de entrada e saída.
- Explicação da solução: descrever como o problema deve ser solucionado. Para soluções por programação dinâmica, deve-se descrever:
 1. estratégia de divisão e conquista dos subproblemas
 2. solução recursiva para o problema
 3. subestrutura ótima
- Para soluções por Algoritmos Gulosos:
 1. Os mesmos itens de programação dinâmica
 2. A descrição da escolha óbvia
- Implementação: a implementação deverá ser entregue em um arquivo por problema.
- Análise assintótica: fazer a análise assintótica da melhor solução para o pior caso.

2 Critérios de avaliação

O seguintes critérios serão observados durante a correção:

- Corretude da explicação do problema e da solução proposta eficiente;
- Corretude da análise algorítmica;
- Eficiência da solução encontrada;
- Implementação correta do algoritmo

Os trabalhos deverão ser feitos individualmente e todos os alunos deverão entregar o relatório.

A entrega deverá ser feita até dia **17/11/2020** via moodle.

3 Problemas

Nesta seção são apresentados os problemas propostos.

3.1 Maior sub-matriz quadrada de 1's em uma matriz binária

Considere uma matriz abaixo:

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Deve-se encontrar o **tamanho** da maior sub-matriz quadrada de 1's. Nesse caso, a resposta deverá ser 3 ($M[5, 1]$ até $M[7, 3]$). Caso a célula $M[8, 3]$ fosse 1, o resultado seria 4 ($M[5, 1]$ até $M[8, 4]$).

3.2 Strings binárias sem 1's consecutivos

Dado um número inteiro N , encontrar todas as strings binárias de tamanho N que não contém 1's consecutivos. Exemplo:

Considere $N = 4$

Nesse caso, temos as seguintes combinações:

0000, 0001, 0010, 0100, 1000, 0101, 1010, 1001

Portanto, o resultado final deverá ser 8.

3.3 Subsequência se soma máxima sem elementos adjacentes

Dado um vetor de inteiros V , encontrar a maior soma de uma subsequência de elementos não adjacentes. Exemplo:

Considere $V = \{1, 2, 9, 4, 5, 0, 4, 11, 6\}$

A saída deverá ser: 26, pois os elementos $\{1, 9, 5, 11\}$ produzem a maior soma.

3.4 Contagem de subconjuntos com soma X

Dado um vetor arr de tamanho N e um inteiro X , a tarefa é encontrar a quantidade de subconjuntos de arr que somam X .

Por exemplo, se $arr = \{1, 2, 3, 3\}$ e $X = 6$, a saída deverá ser 3, pois temos 3 subconjuntos que somam 6: $\{1, 2, 3\}$, $\{1, 2, 3\}$ e $\{3, 3\}$.

Por exemplo, se $arr = \{1, 1, 1, 1\}$ e $X = 4$, a saída deverá ser 1, pois só temos 1 subconjunto de arr para que a soma de seus elementos seja 4.

3.5 Subsequência de maior soma

No problema de Subsequência contínua de maior soma o objetivo é encontrar o valor da maior soma de uma subsequência contínua dado um vetor V .

Por exemplo, o vetor $V = \{-2, -3, 4, -1, -2, 1, 5, -3\}$ tem como valor da maior soma de subsequência contínua igual a 7, pois os elementos que compõem essa soma são: $\{4, -1, -2, 1, 5\}$.

3.6 Maior subsequência para um palíndromo

Um palíndromo é uma sequência de caracteres que, quando escrita ao contrário resulta na mesma sequência. Um exemplo de palíndromo é a palavra "radar". Note que, ao ler a palavra de trás para frente, temos a mesma palavra.

O objetivo do problema da *maior subsequência para um palíndromo* é encontrar uma subsequência de letras a partir de uma string S de entrada de tal forma que forme o maior palíndromo possível. Note que uma subsequência é uma sequência derivada de outra através da remoção ou não de alguns elementos, sem alterar a ordem em que aparecem na sequência original. A saída do seu programa deverá ser o tamanho da maior subsequência. Exemplo: $S = (abdbca)$

A saída deverá ser **5**, uma vez que a maior subsequência que forma um palíndromo é $(abdba)$.