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 a subestrutura ótima, a estratégia de divisão e conquista dos subproblemas. Para soluções por Algoritmos Gulosos, deverá ser descrita a solução por programação dinâmica e a escolha óbvia deverá ser concluída dessa descrição.
- Implementação: a implementação deverá ser entregue em um arquivo por problema.
- Análise assintótica: fazer a análise assintótica da melhor solução.

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 poderão ser feitos em duplas e todos os alunos deverão entregar o relatório. No caso de trabalhos feitos em duplas, informar os nomes dos dois componentes no trabalho.

A entrega deverá ser feita via moodle até dia **24/05/2020** via moodle.

3 Problemas

3.1 Fibonacci

A sequência de Fibonacci é muito conhecida na matemática por ser a base de cálculo da razão áurea. Ela é formada a partir de dois números iniciais, 0 e 1, e todos os número da sequência seguem a equação recursiva:

$$Fib(x) = Fib(x-1) + Fib(x-1) \tag{1}$$

Assim, a sequência de Fibonacci é $0, 1, 1, 2, 3, 5, 8, 13, \ldots$. O problema de Fibonacci é encontrar o n-ésimo termo da sequência.

Por exemplo, para o termo $i = 4$ temos o valor 3.

3.2 Troco de moedas

Dado um valor N, se queremos encontrar um trocado em moedas para N – e supondo que temos quantidades infinitas de moedas de $S = \{S_1, S_2, S_3, \ldots, S_m\}$ valores – quantas maneiras diferentes nós podemos trocar esse valor ? A ordem das moedas é importante.

Por exemplo, para $N = 4$ e $S = \{1, 2, 3\}$, há 4 soluções: $\{1, 1, 1, 1\}, \{1, 1, 2\}, \{2, 2\}, \{1, 3\}$. Sendo assim, a saída deve ser 4. Para $N = 10$ e $S = \{2, 5, 3, 6\}$, há 4 soluções: $\{2, 2, 2, 2, 2\}, \{2, 2, 3, 3\}, \{2, 2, 6\}, \{2, 3, 5\}$ e $\{5, 5\}$. Assim, a saída do algoritmo deverá ser 5.

3.3 Menor quantidade de moedas para um valor

Dado um valor V, se queremos dar um troco no valor de V reais – e supondo que temos quantidade infinitas de moedas de $C = \{C_1, C_2, C_3, \ldots, C_m\}$ valores distintos – qual é o menor número de moedas para o valor V?

Por exemplo, para $C = \{25, 10, 4\}$ e $V = 30$ podemos utilizar uma moeda de 25 e outra de 5 centavos. Logo, a saída será 2.

Por exemplo, para $C = \{9, 6, 5, 1\}$ e $V = 11$ podemos utilizar uma moeda de 6 e outra de 5 centavos. Logo, a saída será 2.

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 Problema da soma perfeita

Dado um vetor arr de inteiros e um inteiro K, a tarefa é imprimir todos os subconjuntos no qual somam o valor K. Esse problema é semelhante ao anterior. Porém ele imprime todos os subconjuntos.

Por exemplo, se $arr = \{5, 10, 12, 13, 15, 18\}$ e $K = 30$ a saída deverá ser $\{12, 18\}, \{5, 12, 13\}, \{5, 10, 15\}$.

Por exemplo, se $arr = \{1, 2, 3, 4\}$ e $K = 5$ a saída deverá ser $\{2, 3\}, \{1, 4\}$.

3.6 Maior subsequência crescente

O problema da maior subsequência crescente (MSC) tem por objetivo encontrar o tamanho da maior subsequência dada uma sequência na qual todos os elementos da subsequência estão ordenados de forma crescente. Por exemplo, o comprimento MSC para a sequência $\{10, 22, 9, 33, 21, 50, 41, 60, 80\}$ é 6 e a MSC é $\{10, 22, 33, 50, 60, 80\}$.

Por exemplo, para a sequência $\{3, 10, 2, 1, 20\}$ a MSC será 3, composta pelos elementos $\{3, 10, 20\}$.

Por exemplo, para a sequência $\{3, 2\}$ a MSC será 1, composta pelos elemento $\{3\}$ ou elemento$\{2\}$.

Por exemplo, para a sequência $\{50, 3, 10, 7, 40, 80\}$ a MSC será 4, composta pelos elementos $\{3, 7, 30, 80\}$.

3.7 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: $\{4, -1, -2, 1, 5\}$.

3.8 Distância de edição

A problema de distância de edição recebe como entrada duas strings S_1 e S_2 e , utilizando as operações listadas abaixo, retorno a quantidade de operações mínimas necessárias para transformar S_1 em S_2.

- Inserção: insere um carácter
- Remoção: remove um carácter
- Troca: troca um carácter

Por exemplo, suponha que S_1 = "$FLOAMAX''$ e S_2 = "$VOLMAX''$ é 3, uma vez que são necessárias, no mínimo, 3 operações de edição:

- Inserir “L” após o “O”
- Substituir “L” por “V”
- Remover “F”.

3.9 Cadeia de comprimento máximo de pares

O problema da cadeia de comprimento máximo de pares é resolvido a partir de um conjunto de n pares de números. In cada para, o primeiro número é sempre menor que o segundo. Um par (c, d) pode seguir outro par (a, b) se $b < c$. A cadeia pode ser formada dessa maneira. O objetivo é encontrar a maior cadeia de pares.

Por exemplo, dados os pares: $(5, 24), (39, 60), (15, 28), (27, 40), (50, 90)$. A saída será $(5, 24), (27, 40), (50, 90)$.

Por exemplo, dados os pares: $(11, 20), (10, 40), (45, 60), (39, 40)$. A saída será $(11, 20), (39, 40), (45, 60)$.