



	AV1	AV2	AV3
1ª ch		X	
2ª ch			

Curso:	Disciplina: N584 - Projeto e Análise de Algoritmos	Código/Turma: N584 - 16
Professor/a: Napoleão		Data: 06/05/2019
Aluno/a:		Matrícula:

Instruções:

1. A atividade deve ser realizada em equipe de 3 alunos.
2. Implementações iguais estão sujeitas a anulação definitiva.
3. A entrega é impreterivelmente na segunda-feira, dia 06 de maio de 2019, no laboratório (K11).
4. Haverá arguição individual pelo professor sobre o trabalho, compondo a nota do aluno.
5. Exemplos de questões da arguição são dados ao final da descrição do trabalho.
6. O professor fornecerá um arquivo de entrada teste no momento da arguição.
7. A equipe é inteiramente responsável por realizar uma demonstração de seu programa.

Problema. Implementar um algoritmo recursivo utilizando memoization, um algoritmo de programação dinâmica e um algoritmo recursivo utilizando uma escolha gulosa para o problema de parentização da multiplicação de uma cadeia de matrizes (Seção 15.2 do *CLRS – Introduction to Algorithms*, 3ª Edição). O algoritmo guloso deve verificar o ponto de parentização ótimo local, ou seja, deve determinar a quebra da cadeia de matrizes corrente em duas subcadeias de matrizes considerando apenas a quantidade mínima de multiplicações escalares do produto das duas submatrizes resultantes. Como o problema não possui a propriedade gulosa, este algoritmo não necessariamente encontrará a solução ótima. Cada algoritmo deve ler a partir do diretório corrente um arquivo de entrada (entrada.txt) e gerar um arquivo de saída (saida.txt), conforme especificação abaixo. Você deve saber ilustrar a lógica dos seus algoritmos através de exemplos, interpretar suas estruturas de dados e de controle, e discutir sobre suas complexidades.

Entrada. O arquivo de entrada contém várias instâncias do problema. A primeira linha do arquivo contém apenas um inteiro indicando a quantidade de instâncias a serem resolvidas. A seguir, cada linha contém os dados de uma instância. Cada instância se inicia com um inteiro n ($1 \leq n \leq 100$) que indica o tamanho da cadeia de matrizes. A seguir, na mesma linha, há $n + 1$ inteiros que correspondem às dimensões das matrizes: $p_0 p_1 \dots p_n$. ($1 \leq p_i \leq 10$, $i = 0, 1, \dots, n$).

Saída. A saída consiste de tantas linhas quantas instâncias do problema houver. Em cada linha, devem ser impressos: a parentização correspondente da cadeia de matrizes e o número de multiplicações escalares a serem realizadas na solução obtida.

Exemplo de entrada.

```
3
2 2 3 7
3 2 3 7 2
4 8 3 1 10 1
```

Exemplo de saída memoization.

```
(A1A2) 42
(A1 (A2A3)) 54
(A1 (A2 (A3A4))) 37
```

Exemplo de saída dinâmico.

```
(A1A2) 42
(A1 (A2A3)) 54
(A1 (A2 (A3A4))) 37
```

Exemplo de saída guloso.

```
(A1A2) 42
(A1 (A2A3)) 54
((A1A2) (A3A4)) 42
```

P1. Dado o exemplo $A_{1 \times 4} B_{4 \times 2} C_{2 \times 3} D_{3 \times 2}$, mostrar a árvore de chamadas recursivas do algoritmo de memoization a partir da chamada $metodo(p, 1, 4)$.

P2. Dado o exemplo $A_{1 \times 4} B_{4 \times 2} C_{2 \times 3} D_{3 \times 2} E_{2 \times 5} F_{5 \times 3} G_{3 \times 2}$, com as respectivas matrizes, m e s , de quantidade de multiplicações escalares e de ponto de quebra do subproblema.

m	1	2	3	4	5	6	7
1	0	8	14	20	30	45	51
2		0	24	28	?	78	78
3			0	12	32	54	62
4				0	30	48	?
5					0	30	42
6						0	30
7							0

s	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6
2		0	2	2	4	2	2
3			0	3	4	4	4
4				0	4	4	4
5					0	5	6
6						0	6
7							0

a) Calcular o valor de $m[2,5]$.

$$m[2,5] = \min \left\{ \begin{array}{l} \\ \\ \\ \\ \\ \end{array} \right. =$$

b) Indicar a parentização ótima de $A_{2..6}$.

B C D E F

P3. Discutir sobre a inversão de cada *for* do algoritmo de programação dinâmica.

P4. Discutir sobre a complexidade do algoritmo dinâmico utilizando notação assintótica.

P5. Dado o exemplo $A_{1 \times 4} B_{4 \times 2} C_{2 \times 3} D_{3 \times 2} E_{2 \times 5} F_{5 \times 3} G_{3 \times 2}$. Indicar a parentização gulosa de $A_{2..7}$.

B C D E F G

P6. Discutir sobre a complexidade do algoritmo guloso utilizando notação assintótica. Caracterizar melhor e pior casos e calcular suas complexidades.