MAC0338 – Análise de Algoritmos
Segundo Semestre de 2018
Segunda Prova – 23 de outubro

Nome do aluno: ____ **Curso**: ______________

Assinatura: ____

No. USP: _________________________ **Professor**: ___________________________________

Instruções

1. Não destaque as folhas deste caderno.
2. A prova pode ser feita a lápis.
3. A legibilidade também faz parte da nota!
4. A prova consta de cinco questões. Verifique antes de começar a prova se o seu caderno de questões está completo.
5. Não é permitido o uso de folhas avulsas para rascunho.
6. Não é necessário apagar rascunhos no caderno de questão mas especifique qual é a resposta e qual é o rascunho.
7. A prova é sem consulta e tem duração de duas horas.
8. Você pode usar, sem ter que escrever, algoritmos vistos em aula. Neste caso, antes de usá-lo, deixe claro qual é o protótipo do algoritmo, o que ele devolve/faz, e quanto tempo ele consome em função de sua entrada.

Não escrever nesta parte da folha

Questão	Vale	Nota
1	2,5	
2	2,5	
3	2,5	
4	2,5	
Total		

Boa prova!

1. [2,5 pontos]
   Considere o problema, visto em aula, de encontrar o número mínimo de multiplicações escalares necessário para calcular um produto $A_1 A_2 \cdots A_n$ de n matrizes. Lembre-se que o algoritmo visto recebe como entrada um vetor $p[0 \,.\,.\, n]$ de inteiros e um inteiro n, todos positivos, e que cada matriz A_i tem $p[i-1]$ linhas e $p[i]$ colunas.

   (a) Escreva o pseudocódigo de um algoritmo de programação dinâmica que recebe p e n como acima e devolve o número mínimo de multiplicações escalares necessárias **e também algum dado extra** (por exemplo, um vetor ou matriz), que permita recuperar uma parentização ótima para multiplicar as matrizes.

   (b) Escreva o pseudocódigo de um algoritmo que recebe p e n como acima, assim como o certo **dado extra**, e imprima uma parentização ótima para multiplicar as n matrizes. A saída deve conter exatamente $n-1$ pares de parênteses. Por exemplo, seu algoritmo pode imprimir $((A_1 A_2)(A_3((A_4 A_5)A_6)))$.

2. [2,5 pontos]
Suponha que temos n notas (de dinheiro) ou moedas com denominações em $v[1\,..\,n]$ em ordem crescente; por exemplo, desconsiderando nossas moedas de centavos, teríamos $v[1] = 1$, $v[2] = 2$, $v[3] = 5$, $\ldots$, $v[7] = 100$. Temos também um vetor $q[1\,..\,n]$ que guarda a quantidade que temos de cada nota ou moeda em caixa. Escreva o pseudocódigo de um algoritmo de programação dinâmica que recebe os vetores v e q e inteiros $n > 0$ e $m > 0$ e calcula quantas formas existem para formar um troco de valor m utilizando nosso suprimento disponível de notas e moedas.

Por exemplo, para a entrada com $n = 7$ descrita como

	1	2	3	4	5	6	7
v	1	2	5	10	20	50	100
q	3	2	1	0	2	1	2

e $m = 104$, existem 2 maneiras de formar o valor m: como $104 = 100 + 2 + 2$ e como $104 = 100 + 2 + 1 + 1$.

Justifique a corretude e o consumo de tempo do seu algoritmo.

3. [2,5 pontos]
   Descreva um algoritmo que recebe como entrada um vetor $x[1\,..\,n]$ de números e um inteiro *par* $n > 0$ e particiona os n números em $n/2$ pares da seguinte forma. Denote por $s_1, s_2, \ldots, s_{n/2}$ as $n/2$ somas dos pares. O algoritmo deve encontrar uma partição em pares que minimiza $\max_i s_i$, ou seja, que minimiza a soma máxima. O consumo de tempo do seu algoritmo deve ser $O(n \log n)$. **Justifique** a corretude e o consumo de tempo do seu algoritmo.

4. [2,5 pontos]
   Seja G um grafo conexo com custo c_e para cada aresta e. Fixe uma aresta f de G. Prove que f está em alguma MST do grafo se, e somente se, f tem custo mínimo dentre todas as arestas de algum corte de G.