MAC 5711 – Análise de Algoritmos
SEGUNDO SEMESTRE DE 2005
Segunda Prova – 18 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 4 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.

Não escrever nesta parte da folha

Questão	Nota	Observação
1		
2		
3		
4		
Total		

Boa Sorte !

1. [3,5 pontos]
   Escreva um algoritmo de complexidade $O(n \lg k)$ para intercalar k listas ordenadas numa só lista ordenada, onde n é o número total de elementos de todas as listas. Mais precisamente, o algoritmo deve ter a seguinte entrada e saída:

   - *Entrada:* Inteiros n e k e vetores ordenados $A_1, \ldots, A_k$.
   - *Saída:* Vetor ordenado B contendo todos os elementos dos A_i's.

   Explique sucintamente porque o seu algoritmo funciona e porque tem complexidade $O(n \lg k)$.

   É possível fazer um algoritmo com complexidade $o(n \lg k)$ para esse problema? Justifique a sua resposta. (*Dica:* Pense em $k = n$.)

2. [3,0 pontos]
Digamos que um *ponto* é um certo tipo de objeto (a natureza exata desses objetos é irrelevante). Suponha que f é uma função que leva ternos ordenados de pontos em números racionais. Suponha que $x_1, \ldots, x_n$ é uma seqüência de pontos tal que $f(x_1, x_2, x_i) \geq 0$ para todo $i \geq 3$. O seguinte algoritmo recebe $x_1, \ldots, x_n$, com $n \geq 2$, e devolve uma subseqüência $p_1, \ldots, p_t$:

ALGO$(n, x_1, \ldots, x_n)$
$p_1 \leftarrow x_1$
$p_2 \leftarrow x_2$
$t \leftarrow 2$
para $i \leftarrow 3$ **até** n **faça**
    enquanto $f(p_{t-1}, p_t, x_i) < 0$ **faça**
        $t \leftarrow t - 1$
    $p_{t+1} \leftarrow x_i$
    $t \leftarrow t + 1$
devolva $p_1, \ldots, p_t$

Mostre que o algoritmo consome tempo $O(n)$ no pior caso.

3. [3,5 pontos]
   Seja $1, \ldots, n$ um conjunto de *tarefas*. Cada tarefa consome um dia de trabalho; durante um dia de trabalho somente uma das tarefas pode ser executada. Os dias de trabalho são numerados de 1 a n. A cada tarefa t está associado um *prazo* p_t: a tarefa deveria ser executada em algum dia do intervalo $1 \,..\, p_t$. A cada tarefa t está associada uma *multa* não-negativa m_t. Se uma dada tarefa t é executada depois do prazo p_t, sou obrigado a pagar a multa m_t (mas a multa não depende do número de dias de atraso).

   Problema: Programar as tarefas (ou seja, estabelecer uma bijeção entre as tarefas e os dias de trabalho) de modo a minimizar a multa total.

   Escreva um algoritmo guloso para resolver o problema. Prove que seu algoritmo está correto. Analise o consumo de tempo.

4. [3,0 pontos]
   Suponha que o padrão P pode conter ocorrências de um caracter *vão* $\diamondsuit$, que pode ser casado a uma cadeia arbitrária de caracteres (inclusive com a cadeia vazia). Por exemplo, o padrão $ab\diamondsuit ba\diamondsuit c$ ocorre no texto $cabccbacbacab$ de duas maneiras diferentes: $c\mathbf{ab}cc\mathbf{ba}cba\mathbf{c}ab$ e $c\mathbf{ab}ccbac\mathbf{bac}ab$. Note que o caracter vão pode ocorrer um número arbitrário de vezes no padrão, mas não ocorre no texto nenhuma vez. Descreva um algoritmo para determinar se um tal padrão ocorre em um texto. Seu algoritmo deve consumir tempo $O(m+n)$, onde m é o número de caracteres no padrão e n é o número de caracteres no texto. Explique porque o seu algoritmo resolve o problema e porque tem a complexidade pedida.