PRIMEIRA PROVA DE PRINCÍPIOS DE DESENVOLVIMENTO DE ALGORITMOS
BCC, 2o. SEMESTRE DE 2006

Instruções:

1. Não destaque as folhas do caderno de soluções.
2. A prova pode ser feita a lápis. Cuidado com a legibilidade.
3. Não é permitido o uso de folhas avulsas para rascunho.
4. Não é necessário apagar rascunhos no caderno de soluções.
5. Asserções imprecisas valem pouco. Justifique suas asserções (dentro do razoável!).
6. **IMPORTANTE: Faça no máximo 3 questões.**

1. [2 pontos] Diga se são verdadeiras ou falsas as seguintes fórmulas:
   (i) $1+2+\cdots+n=\sum_{1\le k\le n} k = O(n^2)$
   (ii) $1^5+2^5+\cdots+n^5=\sum_{1\le k\le n} k^5 = O(n^6)$
   (iii) $1+2+2^2+\cdots+2^n=\sum_{0\le k\le n} 2^k = O(2^n)$
   (iv) $2^{S_n}=O(2^{2^n})$, onde $S_n=\sum_{0\le k\le n} 2^k$ é como em (iii)
   Procure justificar suas respostas.

2. [3 pontos] Considere o problema da conexidade, estudado no começo do semestre: são dados pares $p_1,\dots,p_M$ com $p_i=(x_i,y_i)$ e $0\le x_i,y_i<N$ para todo i. Queremos 'filtrar' esta seqüência de pares de forma que apenas aqueles pares 'essenciais' sejam impressos (o par p_i deve ser impresso se e só se x_i e y_i pertencem a diferentes componentes conexas definidas pelos pares p_j $(j<i)$).
   (i) Suponha que a entrada é
   $$(3,4),(4,9),(8,0),(2,3),(5,6),(5,9),(7,3),(4,8),(6,1).$$
   Qual é a saída correspondente? Não dê apenas a saída; mostre como você chegou a ela (basta fazer uns diagramas e explicar o que está "acontecendo").
   (ii) Descreva a solução que vimos para este problema que consome tempo $O(M\log N)$. Não é necessário escrever o código C correspondente a sua solução, bastando descrever a estrutura de dados e os algoritmos usados (mas faça isso de forma bem precisa).
   (iii) Por que esta solução consome tempo $O(M\log N)$? Argumente da forma mais precisa que você conseguir.

3. [3 pontos] Escreva um algoritmo para resolver o seguinte problema. Dada uma cadeia de caracteres, determinar o número de ocorrências de *números naturais* na cadeia.

   Um número natural deve ser entendido como uma cadeia não-vazia maximal de dígitos, sem sinal. Isto é, um número natural é uma cadeia não-vazia de dígitos (`'0'..'9'`) que não pode ser estendida nem à esquerda nem à direita. Por exemplo, com a entrada

Pergunte-me 123 e responderei 456. Tão simples como 789 só 10.

seu algoritmo deve devolver 4.

Escreva sua solução na forma de pseudocódigo (isto é, não é necessário escrever em C). A estrutura geral de sua solução deve ser como segue

Algoritmo Numero_de_naturais(s)
1. ...
2. **enquanto** há caracteres para ler
3. $c =$ próximo caracter de s
4. ...
5. **devolva** cont

4. [4 pontos] Nesta e na próxima questão, você deve escrever código em C. Ademais, supomos que temos

```
typedef struct node *link;
struct node { int item; link next; };
```

Escreva uma função de protótipo

```
link elem_max(link h);
```

que recebe como entrada um ponteiro para a cabeça de lista `heada` de uma lista, e devolve (um ponteiro para) uma célula desta lista que tem campo `item` máximo. Ademais, a célula que é devolvida por `elem_max()` deve ser removida da lista dada. Supomos que temos uma cabeça de lista mas não temos cauda em nossa lista. Se a lista dada for vazia (isto é, se `h->next == NULL`), a função deve devolver `NULL`.

O que ocorreria se não tivéssemos cabeça de lista nesta questão?

5. [2 pontos] Escreva uma função de protótipo

```
link ordene(link h);
```

que recebe como entrada um ponteiro para a cabeça de lista `heada` de uma lista, e devolve um ponteiro para uma nova lista constituída pelas células na lista original, mas em ordem não-decrescente. A lista que você criar deve ter uma cabeça de lista. Use a função `elem_max()` da questão anterior, mesmo que você não a tenha feito.