Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Sistemas de Informação

IF969 Algoritmos e Estruturas de Dados — 2012.2
Prof. Paulo Fonseca

TERCEIRA LISTA DE EXERCÍCIOS
03 de Abril de 2013

- Esta lista contém **04** questões escritas e **02** questões práticas de programação.
- As respostas às questões escritas devem ser entregues ao professor em papel escrito à mão (à tinta e legível) até o final da aula do dia **17/Abr/13**.
- As questões práticas devem ser submetidas através do sistema de correção automática em `www.cin.ufpe.br/~if969`, seção "Listas" até o dia **16/Abr/13 às 17h** (hora do servidor)
- A lista é **individual**.
- É permitido discutir as questões com os colegas e monitores mas as respostas devem ser escritas individualmente com suas próprias palavras.
- É permitido estudar códigos de terceiros mas *não é* permitido copiar trechos substanciais e apresentá-los como trabalho seu.
- A detecção de cópia implicará na atribuição de nota 0 (zero) à lista.
- O professor poderá solicitar aleatoriamente a explicação oral de suas respostas até a divulgação das notas. Essa explicação deverá ser feita sem assistência. Esteja certo de ter compreendido suas respostas.

QUESTÃO 1 (1,5 pts) Considere o vetor $V = (1, 3, 6, 5, 9, 7, 2, 8, 4)$. Represente:

a) A construção da *max-heap* binária resultante das inserções sucessivas dos elementos de V (na mesma ordem em que aparecem no vetor);
b) A construção bottom-up da *max-heap* binária com a função *max-heapify*.

QUESTÃO 2 (1,5 pts) Reescreva de maneira não-recursiva o algoritmo $find$ para conjuntos disjuntos usando a heurística da compressão de caminhos.

Algoritmo *find_PC*
Entrada Um elemento (nó) x
Saída A raiz da classe de equiv. $[x]$
1 **se** $x \rightarrow father \neq x$ **então**
2 $\quad x \rightarrow father \leftarrow$ *find_PC*$(x \rightarrow father)$
3 **fim se**
4 **devolva** $x \rightarrow father$
fim

QUESTÃO 3 (1,5 pts) Suponha que queiramos procurar palavras numa lista encadeada com n palavras. Suponha que o tamanho de cada palavra é $\leq m$, para um $m > 0$ dado.

a) Qual o custo, no pior caso, de procurar k palavras?
b) Explique como podemos melhorar nossa procura usando uma tabela de dispersão (*hashtable*), indicando o custo da procura das k palavras neste caso.

QUESTÃO 4 (1,5 pts) Reescreva o algoritmo do percurso em profundidade visto em aula usando uma pilha para eliminar a recursão.

QUESTÕES 5 e 6 (2 pts cada)

Questões prática de implementação. Ver Lista 3 em `www.cin.ufpe.br/~if969`.

EOF