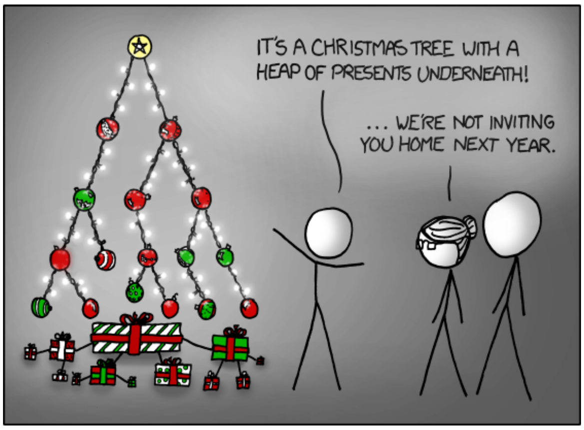
Universidade Federal de Pernambuco
Centro de Informática
Bacharelado em Ciência da Computação

IF672cc Algoritmos e Estr. de Dados — 2013.2
Profs. Katia Guimarães, Paulo Fonseca

PRIMEIRA PROVA — 19 de Dezembro de 2013

- Esta prova contém 04 (quatro) questões.
- A duração da prova é de 2h00.



QUESTÃO 1 (2,5 pts)

Calcule e exiba a ordem exata de complexidade assintótica do algoritmo a seguir:

Algoritmo *quantocusto*
Entrada $n \in \mathbb{N}$
Saída s

$s \leftarrow 0$
para $i = 1, \ldots, n$ **faça**
    para $j = 1, \ldots, i$ **faça**
        para $k = 1, \ldots, j$ **faça**
            $s \leftarrow s + 1$
        fim faça
    fim faça
fim faça
devolva s
fim

Dica: $\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$ e $\sum_{i=1}^{n} i^2 = \frac{1}{6}n(n+1)(2n+1)$.

QUESTÃO 2 (2,5 pts)

O *diâmetro* (ou *largura*) de uma árvore binária T, $D(T)$, é definido como o comprimento (número de nós) do maior caminho entre duas folhas de T.

a) Escreva a definição recursiva de $D(T)$. *Dica:* Se o caminho máximo não passa pela raiz, ele está totalmente contido em uma das sub-árvores à direita ou à esquerda da raiz. Se ele passa pela raiz, o seu comprimento pode ser calculado a partir das alturas dessas sub-árvores.

b) Escreva em pseudocódigo um algoritmo que recebe como entrada um ponteiro para a raiz de uma árvore binária e retorna o seu diâmetro.

QUESTÃO 3 (2,5 pts)

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

a) A max-heap binária resultante das inserções sucessivas dos elementos de V (na mesma ordem em que aparecem no vetor);

b) A max-heap binária resultante da construção *bottom-up* com a função *max-heapify*.

QUESTÃO 4 (2,5 pts)

A estrutura de dados de floresta para conjuntos disjuntos sobre um universo $\mathcal{A} = \{a_1, \ldots, a_n\}$ pode ser representada por um array `P` $= (\texttt{p}[1], \ldots, \texttt{p}[n])$, onde $\texttt{p}[i] = j$ sse a_j for o 'pai' do elemento a_i (para as raízes, $\texttt{p}[i] = i$).

Considere uma partição inicial P_0 de $\mathcal{A}$ na qual cada elemento pertence a uma classe de equivalência unitária formada por ele apenas, isto é, $P_0 = \{\{1\}, \ldots, \{n\}\}$. Complete o diagrama abaixo para obter a configuração de `P` ao final das operações indicadas, assumindo as heurísticas de *união ponderada* com *compressão de caminhos*. *Lembrete:* `Union`(i, j) é implementado como `Link`$($`Find`(i), `Find`$(j))$. Em caso de empate, `Link` deve escolher o representante do primeiro conjunto para representante da união.

P =	1	2	3	4	5	6	7	8	9	10	Início P_0
	1	1	3	4	5	6	7	8	9	10	`Union(1,2)`
											`Union(3,4)`
											`Union(5,6)`
											`Union(7,8)`
											`Union(9,10)`
											`Union(1,6)`
											`Union(8,4)`
											`Union(10,3)`
											`Union(4,6)`