



Primeira Prova — 02 de Maio de 2016

- Esta prova tem 05 questões.
- A duração da prova é de 02h00min.

■ QUESTÃO 1 (2,0pt)

A *amplitude* de um conjunto finito não-vazio $S \subset \mathbb{N}$ é definida como a diferença entre o maior e o menor elementos. Indique a complexidade do *melhor* caso e do *pior* caso do algoritmo ótimo para determinar a amplitude de um conjunto dado de n elementos quando ele é representado por cada uma das estruturas a seguir.

- a) Um array não-ordenado.
- b) Um array ordenado.
- c) Uma lista simplesmente encadeada ordenada.
- d) Uma BST.

■ QUESTÃO 2 (2,0pt)

Considere uma variação do algoritmo Quicksort visto em aula no qual o pivô é escolhido aleatoriamente como o elemento mais à esquerda ou o mais à direita do intervalo a ser particionado. Forneça um vetor de $n = 8$ elementos que corresponda ao *pior* caso do algoritmo. Explique sucintamente (máx. 4 linhas) por que o vetor fornecido é um exemplo de pior caso.

■ QUESTÃO 3 (2,0pt)

A remoção sucessiva de dois valores de uma BST resulta sempre na mesma árvore, independente da ordem na qual forem feitas as duas remoções? Se sim, justifique brevemente (máx. 4 linhas). Se não, forneça um contra-exemplo.

■ QUESTÃO 4 (2,0pt)

Considere o vetor

$$V = (1, 8, 6, 5, 3, 7, 4).$$

- a) Ilustre a construção da *max-heap* através da inserção sucessiva dos elementos de V .
- b) Ilustre a construção *bottom-up* de uma *max-heap* a partir do vetor V .
- c) Os resultados da construção por inserções sucessivas e *bottom-up* dão sempre o mesmo resultado? Se sim, justifique sucintamente (máx. 4 linhas). Se não, dê um contra-exemplo de tamanho *mínimo*.

■ QUESTÃO 5 (2,0pt)

Considere a estrutura *union-find* com as habituais operações $find(x)$ e $union(x, y)$, e acrescida da operação $undo_union()$ que desfaz a última união não-desfeita. Esta estrutura necessita manter um histórico das uniões realizadas. Portanto, para cada união $union(x, y)$

realizada, o par (r_x, r_y) é colocado numa pilha, sendo r_x e r_y os representantes das classes de equivalência de x e y , respectivamente.

- a) Supondo a representação por florestas apenas com a heurística da *união ponderada* (pelas alturas das árvores), e supondo ainda que a estrutura possui, inicialmente, 8 classes unitárias com os elementos A, B, C, \dots, H (todas com altura 1), complete o diagrama a seguir com o resultado das operações indicadas.

#	Operação	Floresta	Pilha
1.	$union(A, B)$		$[(A, B)]$
2.	$union(C, D)$		$[(A, B), (C, D)]$
3.	$union(E, F)$		
4.	$union(G, H)$		
5.	$union(A, C)$		
6.	$union(F, H)$		
7.	$union(D, H)$		
8.	$union(F, G)$		
9.	$undo_union()$		
10.	$undo_union()$		

- b) Supondo agora que a estrutura também utilizasse a heurística da *compressão de caminhos*, qual seria a configuração da floresta e da pilha após a operação #8? Nesse caso, teríamos informação suficiente para realizar as operações #9 e #10? Justifique sucintamente.

