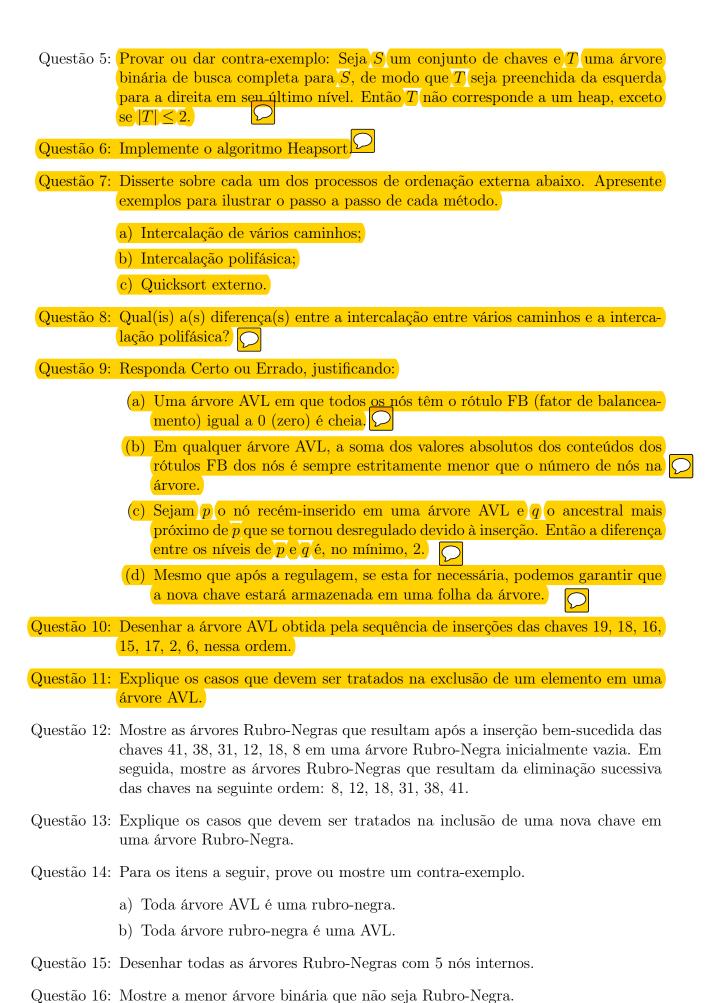
Estrutura de Dados II Lista de Exercícios

Data de entrega: 16/12/2013

Questão 1:	Sobre os algoritmos de ordenação vistos em sala, resolva os itens a seguir. Caso seja solicitado o algoritmo, apresente a implementação em linguagem C.
	a) Implemente o algoritmo quicksort aplicado a sequências de elementos de tipo genérico sem repetição de elementos.
	b) Implemente o algoritmo quicksort que trata elementos de tipo genérico repetidos no vetor.
	c) Crie uma massa de testes grande o suficiente para comparar o tempo de execução das duas versões implementadas acima. Crie testes com muitos elementos repetidos e testes sem elementos repetidos. Monte uma tabela para comparar o tempo de execução de cada algoritmo.
	d) Sobre o Mergesort, implemente a função de intercalação (merge).
	e) Implemente o Mergesort que utiliza a função de intercalação do item anterior,
	f) O que é um algoritmo de ordenação estável?
(g) Mostre que o Mergesort é um algoritmo de ordenação estável.
	h) O algoritmo Quicksort não é um algoritmo de ordenação estável. Que tipo de transformação você poderia fazer nas chaves para que ele se torne um algoritmo estável?
	i) Sobre os algoritmos de ordenação em tempo linear, enumere as características das entradas para cada algoritmo visto em sala.
	j) Implemente em linguagem C, cada algoritmo de ordenação em tempo linear visto em sala.
Questão 2:	Verificar se estas sequências correspondem ou não a um heap.
((a) 33, 32, 28, 31, 26, 29, 25, 30, 27 (b) 33, 32, 28, 31, 29, 26, 25, 30, 27
Questão 3:	Provar ou dar contra-exemplo: Seja S uma sequência de chaves correspondendo a um heap. Sejam s_i , s_j chaves de S tais que $i < j$ e $s_i < s_j$. Então a sequência obtida pela troca de posições de s_i com s_j é também um heap.
Questão 4:	Repetir o exercício anterior, considerando, agora, $i < j$ e $s_i > s_j$.



- Questão 17: Insira em uma árvore Rubro-Negra, itens com as chaves: 4 7 12 15 3 5 14 18 (nesta ordem). Desenhe a árvore resultante da inserção, sendo que uma nova árvore deve ser desenhada quando houver uma rotação ou troca de cores. (Atenção: verifique a necessidade de rotação e/ou troca de cores a cada inserção).
- Questão 18: Sugira uma estrutura de dados (struct) para representar uma árvore Rubro Negra. Considere que a estrutura deve armazenar tanto a chave quanto o conteúdo associado a esta chave.
- Questão 19: Explique os casos que devem ser tratados na exclusão de um elemento em uma árvore Rubro-Negra.
- Questão 20: Implemente (em linguagem C) a árvore AVL.
- Questão 21: Implemente (em linguagem C) a árvore Rubro-Negra.