

Segunda Avaliação a Distância

1. (1,5) Elabore um algoritmo que, recebendo como entrada uma árvore binária T , calcula para cada nó v de T o número de elementos da subárvore de T enraizada em v . Cada nó v de T apontado por um ponteiro pt possui os campos $pt \uparrow .esq$ e $pt \uparrow .dir$, para apontar para os filhos esquerdo e direito, respectivamente.
2. (1,5) Dado um conjunto de chaves $S = \{s_1, s_2, \dots, s_5\}$, em que $s_1 < s_2 < \dots < s_5$, desenhe todas as possíveis árvores binárias de busca para este conjunto de chaves.
3. (1,5) Desenhe todas as árvores AVL possíveis para o conjunto de chaves $S = \{1, 2, 3, 4, 5\}$.
4. (1,5) Prove ou dê contra-exemplo para a seguinte afirmação: Para qualquer conjunto de chaves e qualquer valor $d > 1$, sempre existe uma árvore B de ordem d que armazena estas chaves.
5. (2,0) Execute o método de ordenação por heap (“heapsort”), aplicando-o às seguintes prioridades (nesta ordem): 18, 25, 41, 34, 14, 10, 52, 50, 48. Desenhe as configurações sucessivas da árvore durante o processo de ordenação.
6. (1,0) Assista às aulas sobre tabelas de espalhamento e faça uma dissertação resumida sobre como funcionam os métodos de tratamento de colisões abordados.
7. (1,0) Responda: como é a árvore de Huffman relativa a n frequências iguais? (Suponha que n é da forma $n = 2^k$, isto é, n é uma potência de 2.)