

Semana 8: Ordenação – Heap Sort

Prof. Dr. Juliano Henrique Foleis

Estude com atenção os vídeos e as leituras sugeridas abaixo. Os exercícios servem para ajudar na fixação do conteúdo e foram escolhidos para complementar o material básico apresentado nos vídeos e nas leituras. Quando o exercício pede que crie ou modifique algum algoritmo, sugiro que implemente-o em linguagem C para ver funcionando na prática.

Vídeos

Heap Sort (Ordenação Por Heap) - Parte 1: Árvores Heap e MaxHeapify

Heap Sort (Ordenação Por Heap) - Parte 2: Construção da Max Heap

Heap Sort (Ordenação Por Heap) - Parte 3: Ordenação por Heap

Leitura Sugerida

FEOFILOFF, Paulo. Projeto de Algoritmos em C. Heapsort ([Link](#))

Exercícios

Exercícios dos materiais de leitura sugerida

Exercícios 1.1, 2.1, 2.2, 2.3, 2.5, 2.6, 3.1, 4.7, 5.1, 5.2, 5.3, 5.6 da página do Prof. Feofiloff (Heapsort) ([Link](#))

Exercícios 7.6, 7.7, 7.8 do livro de Szwarcfiter e Markenzon ([Link](#))

Exercícios Complementares

1. A função $\text{int MaxK}(\text{int } *V, \text{int } n, \text{int } k)$ recebe um vetor V (considere-o desordenado) de tamanho n , e um inteiro k . Esta função deve retornar o k -ésimo maior valor distinto do vetor V . O vetor pode ter repetições. Por exemplo, para $k=2$ e $V = [0, 0, 4, 5]$, MaxK retorna 4. Caso o vetor não possua k elementos distintos, retorne o maior valor de todos. Utilize uma heap máxima para implementar MaxK.

2. Execute o HeapSort em um vetor aleatório gerado com a função $\text{int}^* \text{random_vector}(\text{int } n, \text{int } \text{max}, \text{int } \text{seed})$ com $n = 1000, 10000, 100000, 500000$, $\text{max} = n * 100$ e $\text{seed} = 0$. Repita os experimentos com os mesmos valores para n , dessa vez com um vetor já ordenado. Compare com os resultados obtidos na atividade da Semana 2.

Atividade Para Entregar

Não há atividade para entregar.

BONS ESTUDOS!