

Técnicas de programação para Games

Aula03

Ordenação

Professor Mestre: Adilson Lopes Khouri

4 de novembro de 2018

Sumário

Vetores

Ordenação

Cronograma

Aula	Conteúdo
12/04/2018	XP e banco de dados
17/04/2018	Introdução de estruturas de dados
24/04/2018	Arrays / Matrizes e Ordenação
26/04/2018	Recursão
03/05/2018	Lista Ligada
08/05/2018	Pilha, Fila
10/05/2018	Hash
15/05/2018	Árvore Binária
17/05/2018	Heap
22/05/2018	Grafos
24/05/2018	Prova

Vetores

- ▶ Vetores são estruturas de dados simples que permitem acesso direto em posições específicas da memória em tempo constante
- ▶ São úteis para armazenar dados quando sabe-se o tamanho total do conjunto previamente
- ▶ São estruturas unidimensionais que permitem qualquer armazenamento na memória, strings, inteiros, long e objetos.
- ▶ É possível realizar operações sobre vetores usando conceitos de álgebra linear

Vetores

- ▶ Programar com os alunos
- ▶ Mostrar complexidade

Matrizes

- ▶ Matrizes são vetores de vetores, em outras palavra, imagine que cada posição do vetor foi preenchida com um outro vetor
- ▶ Em modelagem estatística usamos para representar variáveis e seus possíveis valores. Em games podemos usar para representar tabuleiros de jogos como: LIFE.

Matrizes

- ▶ Programar com os alunos
- ▶ Mostrar complexidade

Técnicas de Ordenação

- ▶ Ordenação é uma atividade frequente ao programar
- ▶ Os algoritmos usam estratégias de ordenação variadas que mostram diversas técnicas de programação

InsertionSort

- ▶ Vamos começar pelo algoritmo de ordenação: "insertionSort", a estratégia de ordenação é baseada em ordenar uma "mão de cartas".
- ▶ Onde o jogador colocar uma a uma as menores cartas na esquerda até que a mão toda fique ordenada.

Insertion Sort Execution Example

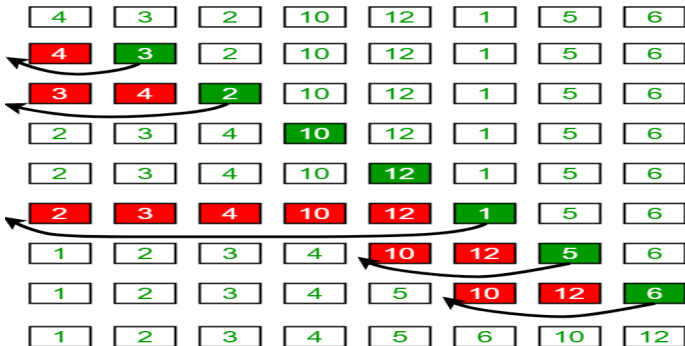


Figura: Exemplo de execução do Insertion Sort [1]

InsertionSort

- ▶ Programar com os alunos
- ▶ Mostrar complexidade

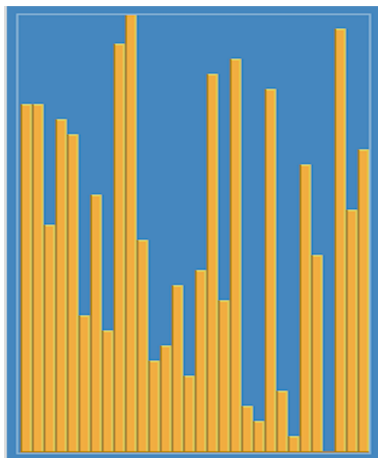


Figura: Exemplo de ordenação com o InsertionSort [1]

BubbleSort

- ▶ Este algoritmo pega cada elemento e troca com o seguinte, contanto que o seguinte seja maior, até o ponto onde não ocorre mais troca. Em seguida, pega o segundo elemento no vetor até acabarem os elementos nos vetores.

BubbleSort

- ▶ Programar com os alunos
- ▶ Mostrar complexidade

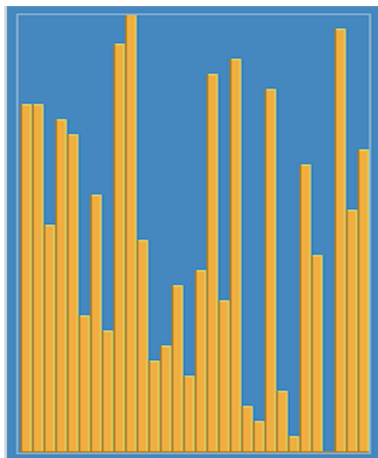


Figura: Exemplo de ordenação com o BubbleSort [1]

MergeSort

- ▶ Particiona o vetor de dois em dois até que só tenha um elemento de forma recursiva. Volta fazendo uma fusão com ordenação dos dois elementos, em seguida, faça a fusão com as outras partes duas a duas até juntar novamente o vetor único e ordenado

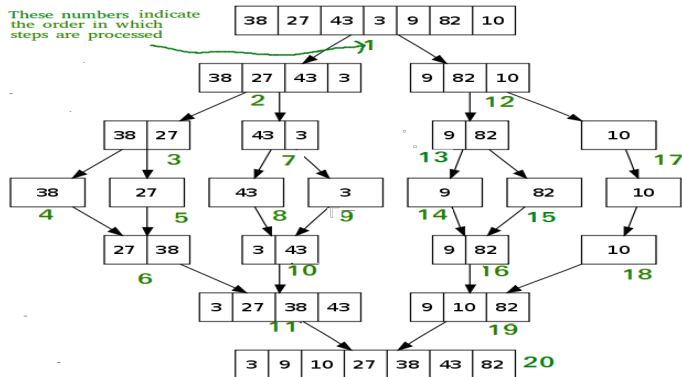


Figura: MergeSort [1]

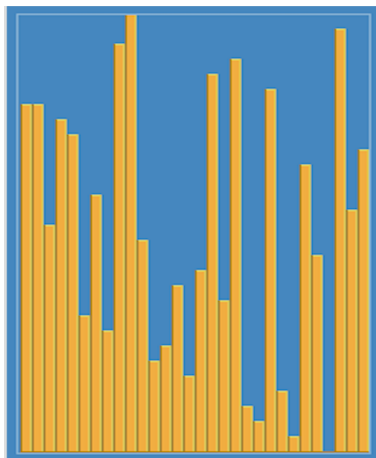


Figura: Exemplo de ordenação com o MergeSort [1]

MergeSort

- ▶ Programar com os alunos
- ▶ Mostrar complexidade

QuickSort

- ▶ Escolhe um pivô no vetor, separa os outros elementos em maiores e menores ou iguais ao pivô. Para cada subconjunto criado (maiores e menores) encontre um novo pivô e repita o procedimento de forma recursiva até que as partições sejam vazias. Ao terminar, o array já está ordenado pois o próprio array foi usado para armazenar as partições.

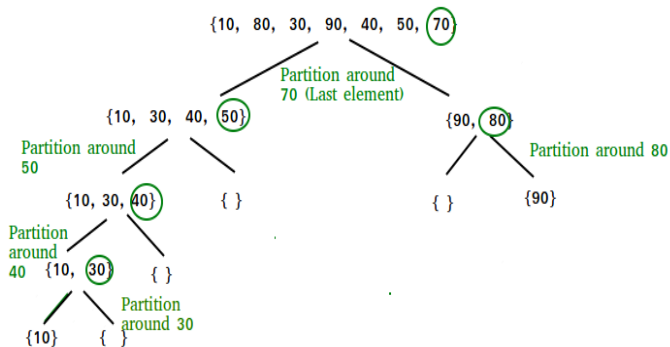


Figura: QuickSort [1]

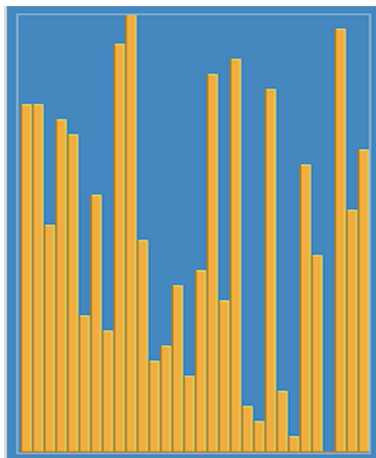


Figura: Exemplo de ordenação com o QuickSort [1]

QuickSort

- ▶ Programar com os alunos
- ▶ Mostrar complexidade

Dúvidas...

Alguma dúvida?

Contato

- ▶ E-mail: *0800dirso@gmail.com* (alunos SENAC)
- ▶ E-mail: *adilson.khoury.usp@gmail.com*
- ▶ Phone: +55119444 – 26191
- ▶ [Linkedin](#)
- ▶ [Lattes](#)
- ▶ [GitHub](#)

Referências

- [1] Geeks. (2018) A computer science portal for geeks. [Online]. Available: <https://www.geeksforgeeks.org>
- [2] A. V. Aho, J. E. Hopcroft, and J. Ullman, *Data Structures and Algorithms*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1983.
- [3] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change (2Nd Edition)*. Addison-Wesley Professional, 2004.
- [4] Beck, *Test Driven Development: By Example*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.
- [5] M. Fowler, *Refactoring: Improving the Design of Existing Code*. Boston, MA, USA: Addison-Wesley, 1999.
- [6] Tutorialspoint. (2017) Tutorialspoint: simple easy learn. [Online]. Available: <https://www.tutorialspoint.com/>