

Técnicas de programação para Games

Aula11

Heap's

Professor Mestre: Adilson Lopes Khouri

4 de novembro de 2018

Sumário

Heaps

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

Heap

- ▶ Estrutura de dados em árvore com algumas propriedades específicas:
- ▶ É uma árvore completa (todos os níveis, com exceção do último tem que ser preenchidos)
- ▶ Um Heap binário sempre será um Min Heap ou Max Heap.
- ▶ Um Heap mínimo tem em sua raiz o menor elemento possível do conjunto. Essa propriedade também se aplica as subárvores do heap.
- ▶ Um Heap máximo tem em sua raiz o maior elemento possível do conjunto. Essa propriedade também se aplica as subárvores do heap.

Heap

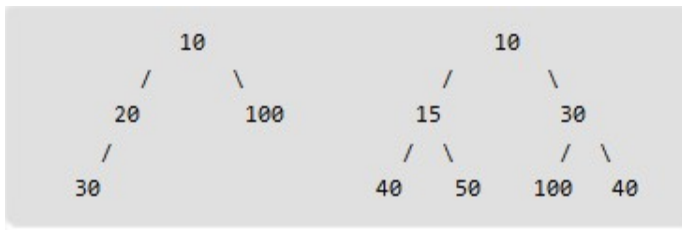


Figura: Exemplo de Heap [1]

Heap

- ▶ Tipicamente Heap's são representados usando vetores (denominado V para exemplo didático) onde:
- ▶ $V[0]$ é a raiz do meu heap
- ▶ Por estar completo em todos os níveis, exceto o último (não sendo obrigatório o preenchimento deste) satisfaz a seguinte relação:
- ▶ $V[1/2]$ retorna o nó pai
- ▶ $V[(2 * i) + 1]$ retorna o filho da esquerda
- ▶ $V[(2 * i) + 2]$ retorna o filho da direita

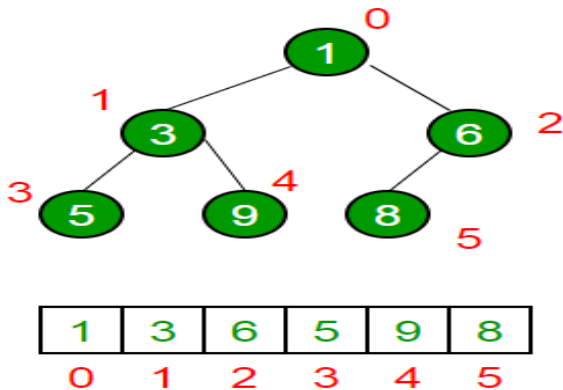


Figura: Exemplo de Heap [1]

Heap

- ▶ As árvores Heap tem algumas aplicações na ciência da computação. Podemos citar o algoritmo de ordenação heapSort, filas de prioridade, grafos pois os algoritmos de Dijkstra e Prim usam essas estruturas.

Heap

- ▶ As operações que podemos realizar em uma estrutura de Heap são:
- ▶ `getRoot` para pegar a raiz de um heap, seja ele min ou max
- ▶ `extractRoot` para remover a raiz do heap, seja ele min ou max
- ▶ `Insert` - sempre mantendo a propriedade de heap
- ▶ `Delete` - sempre mantendo a propriedade de heap

Heap

- ▶ O HeapSort é um algoritmo de ordenação baseado em estruturas heap. Consegue ordenar arranjos em complexidade $O(n \log n)$.
- ▶ A ideia do HeapSort é encontrar o maior elemento e colocar ele na primeira posição, repetindo esse procedimento de forma recursiva para todos os elementos.

Heap

- ▶ O procedimento de ordenação, em alto nível, pode ser visualizado como:
- ▶ Crie um heap máximo com os valores de entrada
- ▶ Nesse ponto o maior valor estará na raiz da árvore
- ▶ Troque ele pelo último valor do heap e decremente a altura do heap em um
- ▶ Repita o procedimento enquanto o tamanho do heap for maior que um

Heap

- ▶ O processo de "heapificar" um nó só pode ser aplicado a um nó cujos filhos estejam heapificados.
- ▶ Dessa forma, devemos aplicar a heapificação de baixo para cima na árvore, começando pelas folhas até chegar na raiz

Exemplo de heapficação

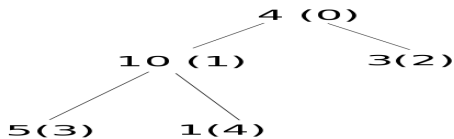


Figura: Exemplo de Heap [1]

Exemplo de heapficação

O processo de heapficação começa pelas folhas: 5, 1, 3. Em nenhum desses casos precisamos alterar a árvore.

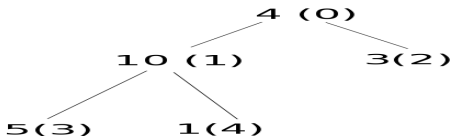


Figura: Exemplo de Heap [1]

Exemplo de heapificação

Ao aplicar a heapificação para a raiz percebemos que há necessidade de trocar valores.

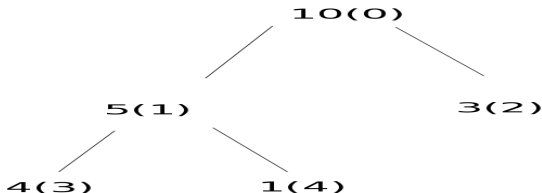


Figura: Exemplo de Heap [1]

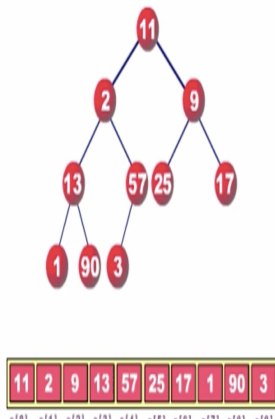


Figura: Exemplo de ordenação com o Heap Sort [1]

Exemplo de uso do HeapSort

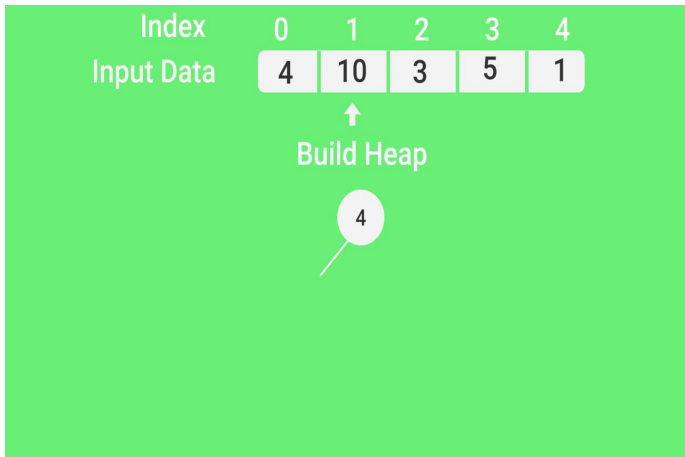


Figura: Exemplo de Heap [1]

Exemplo de uso do HeapSort

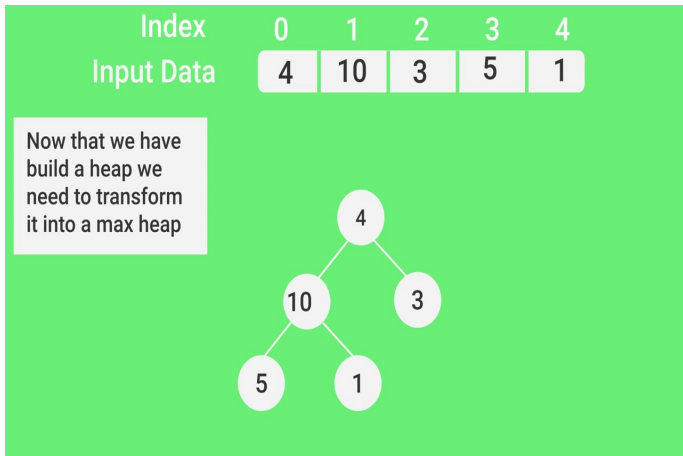


Figura: Exemplo de Heap [1]

Exemplo de uso do HeapSort

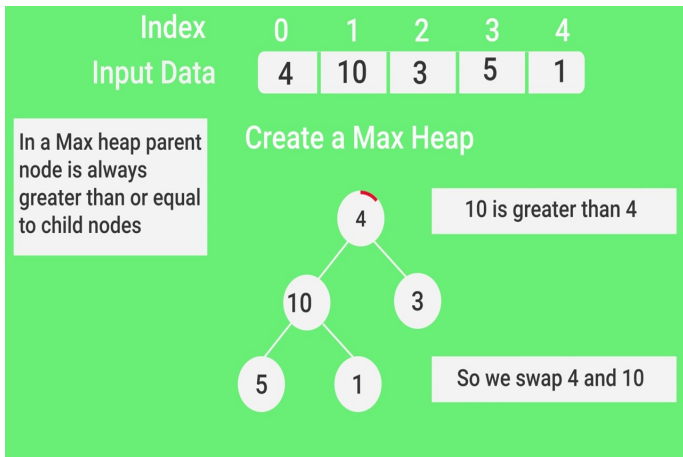


Figura: Exemplo de Heap [1]

Exemplo de uso do HeapSort

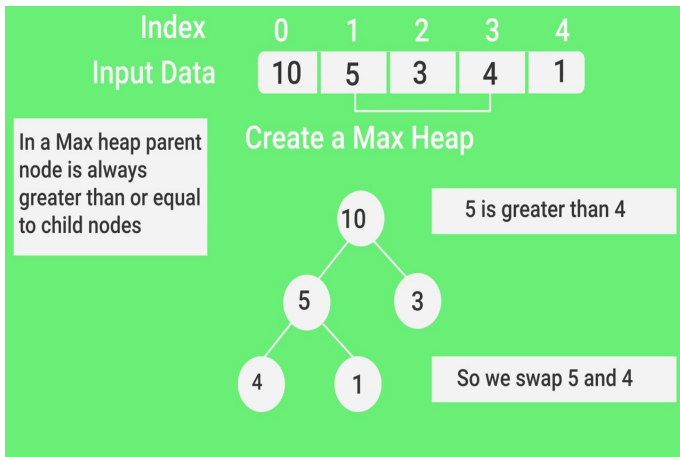


Figura: Exemplo de Heap [1]

Exemplo de uso do HeapSort

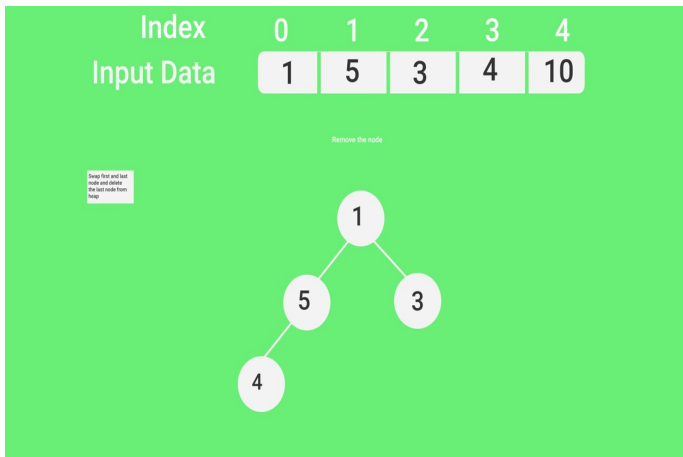


Figura: Exemplo de Heap [1]

Exemplo de uso do HeapSort

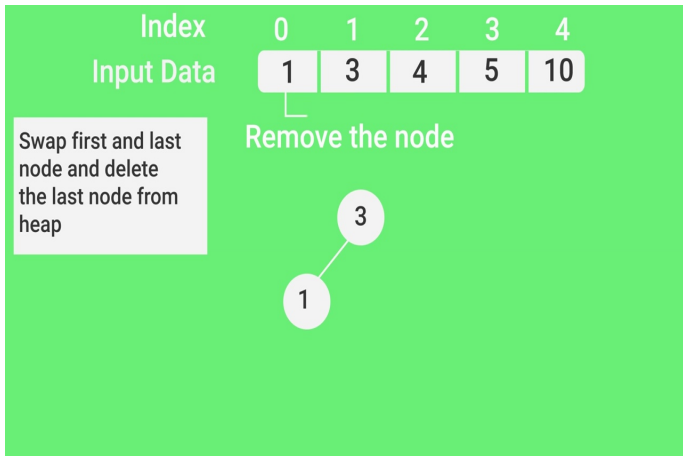


Figura: Exemplo de Heap [1]

Grafos

- ▶ Programar com os alunos em C#
- ▶ Discutir no quadro a comparação com outras estruturas

Dúvidas...

Alguma dúvida?

Contato

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

- [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] Fernanda. (2014) Sql join: Entenda como funciona o retorno dos dados. [Online]. Available: <https://www.devmedia.com.br/sql-join-entenda-como-funciona-o-retorno-dos-dados/31006>
- [7] S. team. (2018) Sqlite sample database. [Online]. Available: <http://www.sqlitetutorial.net/sqlite-sample-database/>