

# Projeto e Análise de Algoritmos

## Exercícios de Heap Sort

Matheus Gabriel

Outubro de 2024

## 1 Exercício 1

### 1.1 Enunciado

**Faltou desenhar a árvore...**

A árvore de decisão para o insertion sort operando sobre 3 elementos. Um nó interno anotado por  $i : j$  indica a comparação entre  $a_i$  e  $a_j$ . Uma folha anotada pela permutação  $\langle \pi_{(1)}, \pi_{(2)}, \dots, \pi_{(n)} \rangle$  indica a ordenação  $a_{\pi_{(1)}} \leq a_{\pi_{(2)}} \leq \dots \leq a_{\pi_{(n)}}$ . O caminho sombreado indica as decisões feitas para ordenar a sequência de entrada  $\langle a_1 = 6, a_2 = 8, a_3 = 5 \rangle$ . A permutação  $\langle 3, 1, 2 \rangle$  na folha indica a sequência ordenada é  $a_3 = 5 \leq a_1 = 6 \leq a_2 = 8$ . Há  $3! = 6$  possíveis permutações dos elementos de entrada, portanto a árvore de decisão deve ter no mínimo 6 folhas.

## 2 O Modelo de Árvore de Decisão

### 2.1 Introdução

Podemos ver algoritmos de ordenação por comparação em termos de árvores de decisão. Uma árvore binária completa que representa as comparações entre os elementos que são feitas por um determinado algoritmo de ordenação operando sobre uma entrada de um determinado tamanho. Controle, movimentação de dados, e todos os outros aspectos do algoritmo são ignorados.

### 2.2 Um limite inferior para o pior caso

O tamanho do caminho mais longo da raiz até qualquer folha em uma árvore de decisão representa o pior caso para o número de comparações que um dado algoritmo efetua. Consequentemente, o pior caso para o número de comparações para um dado algoritmo de ordenação por comparação é igual a altura de sua árvore de decisão.

### Teorema

Qualquer algoritmo de ordenação por comparação requer  $\omega(n \lg n)$  comparações no pior caso.