

CI1056: Algoritmos e Estruturas de Dados II

Prof. Dr. Marcos Castilho

Departamento de Informática/UFPR

7 de dezembro de 2020

Resumo

Limites inferiores para ordenação por comparação entre elementos do conjunto

- Mostrar que o limite inferior para ordenação *por comparação* é $n \cdot \log_2(n)$

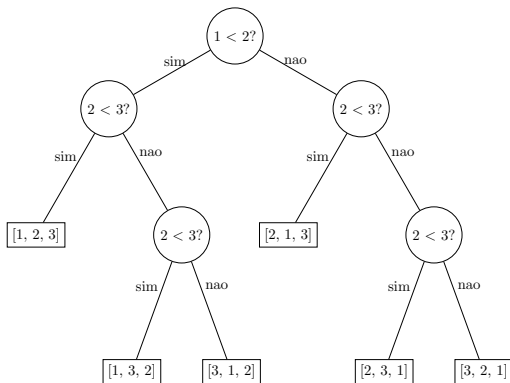
- Será que é possível ordenar com menos de $n \cdot \log_2(n)$ comparações entre os elementos do conjunto?
- Veremos que, no caso geral, não é

Ordenação por comparação entre os elementos do conjunto

- Nós estudamos os custos dos números de comparações entre elementos do conjunto
- Existem outras formas de ordenar
- Mas este é o caso geral
- Veremos que existe um limite inferior para isso

Árvores de decisão

- É uma árvore binária que representa as comparações entre elementos executados por um determinado algoritmo de ordenação aplicado a uma entrada de um tamanho fixo
- Exemplo para 3 elementos:



- Nodos internos: representam as comparações feitas pelo algoritmo
- Nodos folhas: representam a permutação que define a ordenação dos elementos
- Rótulos nas arestas: representam se a comparação retornou verdadeiro ou falso

- Vamos considerar, sem perda de generalidade, que:
 - Todos os elementos do conjunto são diferentes, isto é, $v[i] \neq v[j]$, para $i \neq j$, $1 \leq i, j \leq n$, sendo n o total de elementos do conjunto
 - Todas as comparações podem ser resumidas à $v[i] < v[j]$ (as outras são equivalentes, e não há igualdade)

Execução do algoritmo

- Executar o algoritmo de ordenação corresponde a percorrer um caminho na árvore de decisão da raiz até uma folha
- Cada nodo interno $i < j$? indica que uma comparação foi feita comparando-se os elementos $v[i], v[j]$
- Quando se chega em uma folha o algoritmo estabeleceu a ordenação dos elementos, representada pela respectiva permutação

- Qualquer algoritmo de ordenação correto deve se capaz de produzir cada permutação de sua entrada
- Existem $n!$ permutações de n elementos
- Todas estas permutações devem ser alguma folha na árvore de decisão
- Caso contrário o algoritmo de ordenação não é correto
- Cada uma das folhas deve ser acessível a partir da raiz por um caminho que corresponde a uma execução do algoritmo sendo executado
- Logo, consideraremos árvores de decisão nas quais todas as folhas são acessíveis

Limite inferior para o pior caso

- O comprimento do caminho mais longo desde a raiz até uma folha acessível qualquer representa o número de comparações feitas no pior caso do algoritmo que está sendo executado
- Logo, o número de comparações no pior caso é igual a altura de sua árvore de decisão
- Este é o limite inferior para o pior caso do tempo de execução de qualquer algoritmo de ordenação por comparação

Qualquer algoritmo de ordenação por comparação exige pelo menos um número proporcional a $n \cdot \log_2(n)$ comparações no pior caso

- Basta determinar a altura de uma árvore de decisão na qual cada permutação aparece como uma folha acessível
- Considere uma árvore de decisão de altura h com l folhas acessíveis correspondente a uma ordenação por comparação sobre n elementos
- Cada uma das $n!$ permutações da entrada aparece como alguma folha
- Logo, temos $n! \leq l$
- Lembrando que uma árvore binária de altura h não tem mais do que 2^h folhas, temos

$$n! \leq l \leq 2^h$$

- Retomando:

$$n! \leq l \leq 2^h$$

- Teorema de Stirling, sendo que e é o número neperiano:

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

- Como a função logaritmo é monotonicamente crescente:

$$h \geq \log_2(n)$$

e portanto $h \leq n \cdot \log_2(n)$

- O conteúdo desta aula está no livro Cormen, Leiserson, Rivest e Stein, no capítulo 8, seção 8.1

- Slides feitos em \LaTeX usando beamer
- Licença

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>