QuickSort

Pedro Henrique Del Bianco Hokama
28 de Agosto de 2019

Referências:

- Notas de aulas fortemente baseadas no curso: Stanford Algorithms by Tim Roughgarden
  - `https://www.youtube.com/playlist?list=PLXFMmlk03Dt7Q0xr1PIAriY5623cKiH7V`
  - Vídeos: 5.4 até 7.2
- CLRS: Cap 7
- Cormen T., Desmistificando Algoritmos Cap 3

1 Introdução

O tempo de execução do QuickSort depende da qualidade do pivô escolhido. Um bom pivô é aquele que divide os problemas em partes de tamanho parecido, enquanto um pivô ruim é aquele que deixa duas partes muito desiguais.

Exercício 1.1: Como seria a árvore de recursão e complexidade no pior caso? Por exemplo se você tiver um arranjo já ordenado, e sempre escolhermos o primeiro elemento.

Nesse caso iremos dividir em duas partes, uma vazia e uma com $n-1$ elementos. E o trabalho da partição será pelo menos:

$$n + (n-1) + (n-2) + \ldots + 1 = \Theta(n^2)$$

No melhor caso:

- Se dividirmos o arranjo sempre no máximo, ou seja, na metade.
- Isso aconteceria se encontrássemos a mediana.
- Teremos duas partes com menos que metade dos elementos
- Podemos limitar superiormente pela mesma recorrência do MergeSort
- Sabemos então que no melhor caso QuickSort é $O(n \log n)$

2 Aleatorização do QuickSort

A aleatorização de algoritmos é uma ferramenta importante da bagagem de qualquer profissional da computação. Veremos como aplicar essa técnica no QuickSort e as vantagens que ela pode trazer.

A ideia é escolher é escolher cada pivô aleatoriamente com a mesma probabilidade. E assim escolher um pivô razoavelmente bom, em uma frequência razoavelmente alta.

Um pivô razoavelmente bom seria um que divida o arranjo em 25-75%, que é suficiente para garantir o tempo de execução de $O(n \log n)$

Exercício 2.1: Mostrar a árvore de recursão.

Metade dos elementos resulta numa divisão de 25-75% ou melhor.

2.1 Análise

Teorema 2.1: Para qualquer entrada de comprimento n, o tempo médio de execução do QuickSort (com pivôs aleatórios) é $O(n \log n)$.

- Para demostrar esse teorema precisaremos de lembrar o que é: espaço amostral finito, variáveis aleatórias, esperança, linearidade da esperança.
- Primeiramente considere como entrada um arranjo A de comprimento n.
- Espaço amostral: $\Omega =$ todos as possíveis sequências de escolhas de pivôs no QuickSort.
- Variável aleatória: para qualquer $\sigma \in \Omega$, $C(\sigma) =$ número de comparações entre dois elementos do arranjo feito pelo algoritmo QuickSort dado as escolhas σ. Note que o tempo de execução total do QuickSort é dominado por esse número.
- Notação:
  - $z_i = i$-ésimo menor elemento
  - para uma escolha σ, e $i < j$, seja $X_{ij}(\sigma) =$ número de vezes que z_i e z_j são comparados.

Exercício 2.2: Fixado dois elementos da entrada, quantas vezes eles podem ser comparados?

Podemos então facilmente escrever C em função de X

$$C(\sigma) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij}(\sigma), \qquad \qquad \forall \sigma \in \Omega$$

Pela linearidade da esperança $E[C] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} E[X_{ij}]$.
Como $E[X_{ij}] = \cancel{0.Pr[X_{ij} = 0]} + 1.Pr[X_{ij} = 1] = Pr[X_{ij} = 1]$.
Então $E[C] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr[X_{ij} = 1] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr[z_i \text{ e } z_j \text{ serem comparados}]$.

Teorema 2.2: Para todo $i < j$, $Pr[z_j \text{ e } z_j \text{ serem comparados}] = \frac{2}{(j-i+1)}$

Prova:

- Fixe z_i, z_j com $i < j$.
- Considere o conjunto $\{z_i, z_{i+1}, \ldots, z_{j-1}, z_j\}$.
- Desde que nenhum desses números seja escolhido como pivô, todos serão passados juntos para a mesma chamada recursiva.
- Considere então o primeiro entres $\{z_i, z_{i+1}, \ldots, z_{j-1}, z_j\}$ que é escolhido como pivô:
  - se z_i ou z_j for escolhido primeiro, eles serão escolhidos.
  - se escolher um dos outros então z_i e z_j nunca serão comparados.
- como a escolha é aleatória qualquer um dos elementos do conjunto tem a mesma chance de ser escolhido primeiro (do conjunto) e portando a chance deles serem comparados é

$$2/(j - i + 1)$$

■

Então

$$E[C] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} Pr[z_i \text{ e } z_j \text{ serem comparados}] = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \frac{2}{(j-i+1)}$$

$$E[C] = 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{(j-i+1)}$$

.

Agora note que a soma interna

$$\sum_{j=i+1}^{n}\frac{1}{(j-i+1)} = \frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\ldots+\frac{1}{n-i+1}$$

e ela obtêm a maior quantidade de termos (e o maior valor) quando $i = 1$, então limitamos superiormente a esperança por:

$$E[C] \leq 2n\sum_{k=2}^{n}\frac{1}{k}$$

A soma agora pode ser limitada superiormente por:

$$\sum_{k=2}^{n}\frac{1}{k} \leq \int_{1}^{n}\frac{1}{x}dx = \ln x\Big|_{1}^{n} = \ln n - \cancel{\ln 1} = \ln n$$

.

Portanto:

$$E[C] \leq 2n\ln n$$

e portanto:

O tempo de execução do QuickSort é $O(n\log n)$