

Projeto e Análise de Algoritmos I

Continuação do Selection

Matheus Gabriel

Novembro 2024

Contents

1	Sobre a figura	1
2	Sobre o Select	1
2.1	Operação	2
2.2	Resolvendo uma recorrência	2
3	Radix Sort	2
3.1	Código	2

1 Sobre a figura

Os n elementos são representados por pequenos círculos e cada grupo ocupa uma coluna. As medianas dos grupos são brancas, e a mediana das medianas está identificada como x (quando encontramos a mediana de um número par de elementos, usamos a mediana inferior). São traçadas setas de elementos maiores para elementos menores e, a partir disso, podemos ver que 3 elementos em cada grupo de 5 elementos à direita de x são maiores que x , e 3 em cada grupo de 5 elementos à esquerda de x são menores que x . Os elementos maiores que x são mostrados sob um plano de fundo sombreado.

2 Sobre o Select

O algoritmo **SELECT** determina o i -ésimo menor elem de um vetor de entrada de $n > 1$ elementos, executando as etapas a seguir (se $n = 1$, então SELECT simplesmente retorna seu único valor de entrada como i -ésimo menor).

2.1 Operação

1. Dividir os n elementos do vetor de entrada em $\lfloor \frac{n}{5} \rfloor$ grupos de 5 elementos cada e no máximo um grupo formado pelos $n \bmod 5$ elementos restantes.
2. Encontrar a mediana de cada um dos $\lfloor \frac{n}{5} \rfloor$ grupos, primeiro através da ordenação por inserção dos elementos de cada grupo (há 5 no máximo), e depois escolhendo a mediana de lista ordenada de elementos de grupos.
3. Usar SELECT recursivamente para achar a mediana x das $\lfloor \frac{n}{5} \rfloor$ medianas localizadas na etapa 2 (se houver numero par, x é a mediana inferior).
4. Particionar o vetor de entrada em torno da mediana de medianas x , usando uma versão modificada de **PARTITION**. Seja k uma unidade maior que o número de elementos no lado baixo da partição, de modo que x seja o k -ésimo menor elemento e existam $n - k$ elementos no lado alto da partição.

Se $i = k$, então retornar x . Caso contrário, usar SELECT recursivamente para encontrar o i -ésimo menor elemento no lado baixo se $i \leq k$, ou então o $(i - k)$ -ésimo menor elemento do lado alto, se $i > k$.

2.2 Resolvendo uma recorrência

$$\begin{aligned} T(n) &= \begin{cases} \Theta(1), & \text{se } n \leq 140 \\ T(\lfloor \frac{n}{5} \rfloor) + T(\frac{7n}{10} + 6) + O(n), & \text{se } n > 140 \end{cases} \\ \text{Tempo } O(n) &\text{ no pior caso} \end{aligned}$$

3 Radix Sort

Radix sort é algoritmo usado por máquinas de ordenação de cartão (antigas). Radix ordena resolvendo de modo contra intuitivo, ordenando o dígito menos significativo primeiro.

3.1 Código

Suponha que cada elemento no vetor de n elementos tem d dígitos.

```
RADIX SORT(A, d)
for i ← 1 to d
  do use um algoritmo de ordenação estável para ordenar o vetor A sobre o dígito i
```