“学有所承”——选择问题的比较复杂度研究

研究生：曾钢
导　师：刘兴武副研究员

中国科学院计算技术研究所

2017年6月10日

目录

选择问题的定义

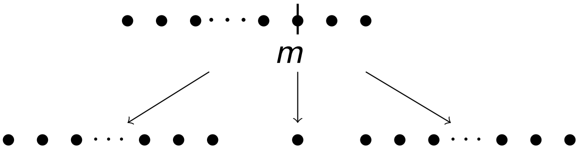
定义 (选择问题)

给定一个由n个元素组成的集合A和一个正整数$k \in [1, n]$，A中所有元素之间均存在大于或小于的关系，即A中所有元素来自于一个全序集。选择问题要求找到A中第k小的元素。

定义 (比较模型)

比较模型限定对数据的访问只可以两两比较，并且算法的复杂度以算法执行过程中的比较次数衡量。称算法的比较复杂度为算法在比较模型中的复杂度。

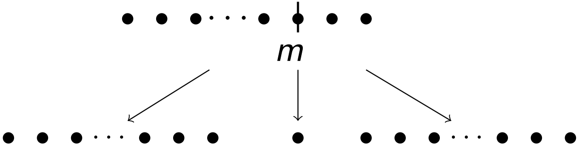
经典算法



$\{x : x \in A\&x < m\}$ $\{m\}$ $\{x : x \in A\&x > m\}$

1. 从A中随机抽取一个元素m

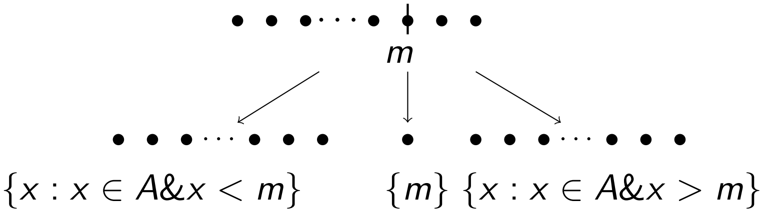
经典算法



$\{x : x \in A\&x < m\}$ $\{m\}$ $\{x : x \in A\&x > m\}$

1. 从A中随机抽取一个元素m
2. 将A划分成三部分：$A_{smaller} = \{x : x \in A\&x < m\}$，$\{m\}$和$A_{larger} = \{x : x \in A\&x > m\}$

经典算法



1. 从A中随机抽取一个元素m
2. 将A划分成三部分：$A_{smaller} = \{x : x \in A \& x < m\}$, $\{m\}$和$A_{larger} = \{x : x \in A \& x > m\}$
3.
$$\begin{cases} \textsc{Select}(A_{smaller}, k) & |A_{smaller}| \geq k \\ \textsc{Select}(A_{larger}, k-1-|A_{smaller}|) & |A_{smaller}| < k-1 \\ m & \text{otherwise} \end{cases}$$

1. 从A中随机抽取一个元素m
2. 将A划分成三部分：$A_{smaller} = \{x : x \in A \& x < m\}$, $\{m\}$和$A_{larger} = \{x : x \in A \& x > m\}$
3. $$\begin{cases} \textsc{Select}(A_{smaller}, k) & |A_{smaller}| \geq k \\ \textsc{Select}(A_{larger}, k-1-|A_{smaller}|) & |A_{smaller}| < k-1 \\ m & \text{otherwise} \end{cases}$$

经典算法

1. 从A中随机抽取一个元素m
2. 将A划分成三部分：$A_{smaller} = \{x : x \in A \& x < m\}$, $\{m\}$和$A_{larger} = \{x : x \in A \& x > m\}$
3. $$\begin{cases} \textsc{Select}(A_{smaller}, k) & |A_{smaller}| \geq k \\ \textsc{Select}(A_{larger}, k-1-|A_{smaller}|) & |A_{smaller}| < k-1 \\ m & \text{otherwise} \end{cases}$$

在最差的情况下，一次迭代仅可删去1个元素，所以有递推式

$$T(n) \leq T(n-1) + O(n),$$

即$T(n) = O(n^2)$

经典算法

Blum et al. (1972)提出了一个选择问题的线性算法

- 随机算法:
  1. 从A中随机抽取一个元素m
  2. 将A划分成三部分：$A_{smaller} = \{x : x \in A \& x < m\}$, $\{m\}$和$A_{larger} = \{x : x \in A \& x > m\}$
  3. $$\begin{cases} \textsc{Select}(A_{smaller}, k) & |A_{smaller}| \geq k \\ \textsc{Select}(A_{larger}, k - 1 - |A_{smaller}|) & |A_{smaller}| < k - 1 \\ m & \text{otherwise} \end{cases}$$

经典算法

Blum et al. (1972)提出了一个选择问题的线性算法

- 随机算法:
  1. 从A中随机抽取一个元素m
  2. 将A划分成三部分：$A_{smaller} = \{x : x \in A \& x < m\}$, $\{m\}$和$A_{larger} = \{x : x \in A \& x > m\}$
  3. $$\begin{cases} \textsc{Select}(A_{smaller}, k) & |A_{smaller}| \geq k \\ \textsc{Select}(A_{larger}, k-1-|A_{smaller}|) & |A_{smaller}| < k-1 \\ m & \text{otherwise} \end{cases}$$
- Blum et al. (1972)的线性算法：
  1. 1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 5$且$|G| = \lceil \frac{n}{5} \rceil$

经典算法

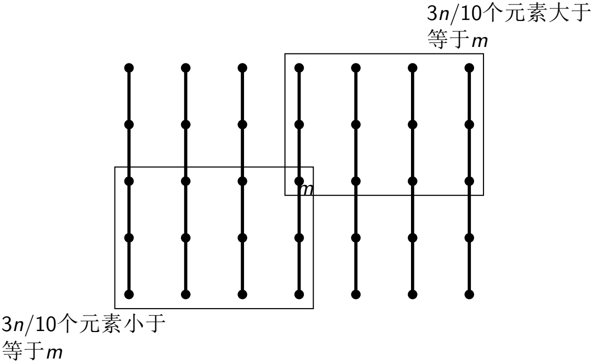
Blum et al. (1972)提出了一个选择问题的线性算法

- 随机算法:
  1. 从A中随机抽取一个元素m
  2. 将A划分成三部分：$A_{smaller} = \{x : x \in A \& x < m\}$, $\{m\}$和$A_{larger} = \{x : x \in A \& x > m\}$
  3. $$\begin{cases} \textsc{Select}(A_{smaller}, k) & |A_{smaller}| \geq k \\ \textsc{Select}(A_{larger}, k-1-|A_{smaller}|) & |A_{smaller}| < k-1 \\ m & \text{otherwise} \end{cases}$$
- Blum et al. (1972)的线性算法：
  1.
     1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 5$且$|G| = \lceil \frac{n}{5} \rceil$
     2. 取集合G中每个块的中位数，构成集合M

Blum et al. (1972)提出了一个选择问题的线性算法

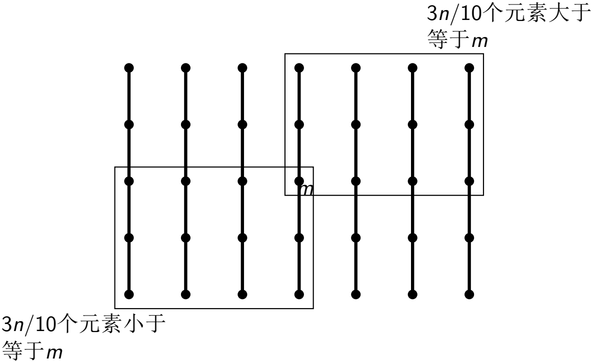
- 随机算法:
  1. 从A中随机抽取一个元素m
  2. 将A划分成三部分：$A_{smaller} = \{x : x \in A \& x < m\}$, $\{m\}$和$A_{larger} = \{x : x \in A \& x > m\}$
  3. $$\begin{cases} \textsc{Select}(A_{smaller}, k) & |A_{smaller}| \geq k \\ \textsc{Select}(A_{larger}, k - 1 - |A_{smaller}|) & |A_{smaller}| < k - 1 \\ m & \text{otherwise} \end{cases}$$
- Blum et al. (1972)的线性算法：
  1.
     1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 5$且$|G| = \lceil \frac{n}{5} \rceil$
     2. 取集合G中每个块的中位数，构成集合M
     3. $m \leftarrow \textsc{Select}(M, \frac{|M|}{2})$
  2. $\cdots$

经典算法



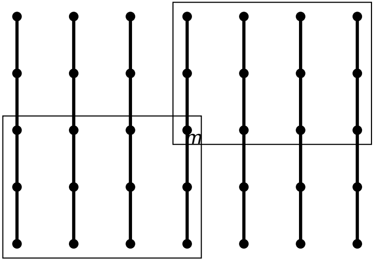
因为m是M的中位数，且M中每个元素为块内的中位数，所以每个元素大于块内的两个元素且小于另外两个元素。

经典算法



因为m是M的中位数，且M中每个元素为块内的中位数，所以每个元素大于块内的两个元素且小于另外两个元素。所以m至少小于A中$\frac{3n}{10}$个元素且至少大于A中$\frac{3n}{10}$，即$T(n) \le T(\frac{n}{5}) + T(\frac{7n}{10}) + O(n)$

经典算法



因为m是M的中位数，且M中每个元素为块内的中位数，所以每个元素大于块内的两个元素且小于另外两个元素。所以m至少小于A中$\frac{3n}{10}$个元素且至少大于A中$\frac{3n}{10}$，即$T(n) \leq T(\frac{n}{5}) + T(\frac{7n}{10}) + O(n)$
因为$\frac{1}{5} + \frac{7}{10} = \frac{9}{10} < 1$，所以$T(n) = O(n)$

研究问题

1.
   1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 5$且$|G| = \lceil \frac{n}{5} \rceil$
   2. 取集合G中每个块的中位数，构成集合M
   3. $m \leftarrow \textsc{Select}(M, \frac{|M|}{2})$
2. $\cdots$

如果改变块的大小，设块的大小为g。假定当g为偶数，每个块的中位数为较小的中位数，即第$\frac{g}{2}$小数。

研究问题

1.
   1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 5$且$|G| = \lceil \frac{n}{5} \rceil$
   2. 取集合G中每个块的中位数，构成集合M
   3. $m \leftarrow \textsc{Select}(M, \frac{|M|}{2})$
2. $\cdots$

如果改变块的大小，设块的大小为g。假定当g为偶数，每个块的中位数为较小的中位数，即第$\frac{g}{2}$小数。
如果g是一个大于4的常数，则有$T(n) = O(n)$

研究问题

1.
   1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 5$且$|G| = \lceil \frac{n}{5} \rceil$
   2. 取集合G中每个块的中位数，构成集合M
   3. $m \leftarrow \textsc{Select}(M, \frac{|M|}{2})$
2. $\cdots$

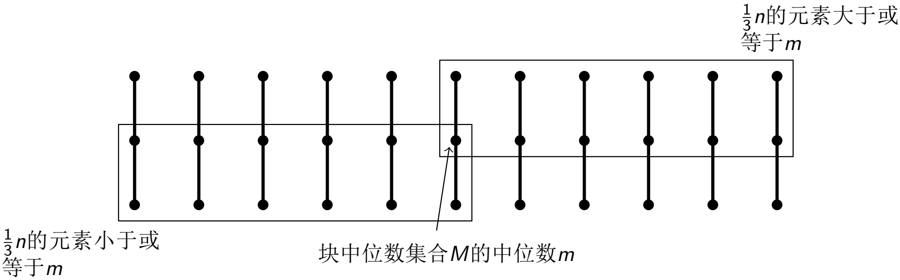
如果改变块的大小，设块的大小为g。假定当g为偶数，每个块的中位数为较小的中位数，即第$\frac{g}{2}$小数。
如果g是一个大于4的常数，则有$T(n) = O(n)$

问题

如果$g = 3$，则算法的比较复杂度是多少？

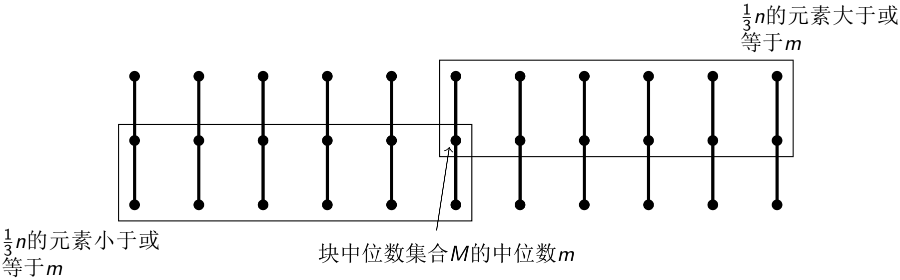
研究问题

$g = 3$



研究问题

$g = 3$

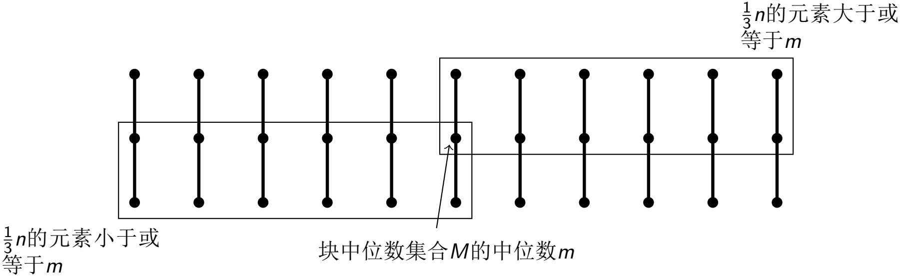


当$g = 3$，有

$$T(n) \leq T(\frac{n}{3}) + T(\frac{2n}{3}) + O(n)$$

研究问题

$g = 3$

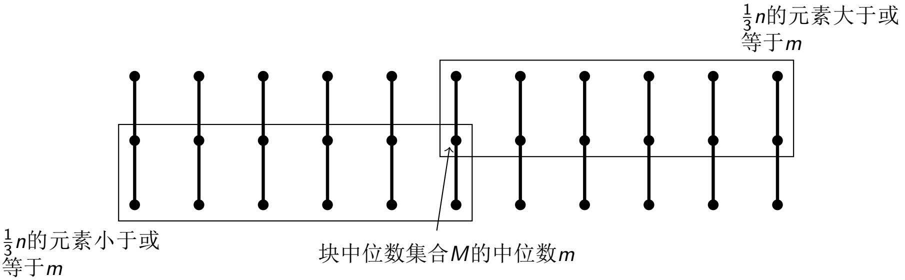


当$g = 3$，有

$$T(n) \leq T(\frac{n}{3}) + T(\frac{2n}{3}) + O(n)$$

所以$T(n) = O(n \log n)$

$g = 3$



当$g = 3$，有

$$T(n) \leq T(\frac{n}{3}) + T(\frac{2n}{3}) + O(n)$$

所以$T(n) = O(n \log n)$
这个分析是不是紧的？目前没有更好的分析，也没有反例证明这个界是紧的。

$\alpha-\beta$算法

Blum等人的算法要求找块中位数集合M的中位数，如果增加算法的表达能力，变为找β分位数，β是根据$\alpha=\frac{k}{n}$调整的值，是否可能可以找到一个策略使得算法比较复杂度为$O(n)$?
算法描述如下

$\alpha-\beta$算法

Blum等人的算法要求找块中位数集合M的中位数，如果增加算法的表达能力，变为找β分位数，β是根据$\alpha=\frac{k}{n}$调整的值，是否可能可以找到一个策略使得算法比较复杂度为$O(n)$?
算法描述如下

- Blum et al. (1972)的线性算法:
    1. 1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 3$且$|G| = \lceil \frac{n}{3} \rceil$
        2. 取集合G中每个块的中位数，构成集合M
        3. $m \leftarrow \textsc{Select}(M, \frac{|M|}{2})$
    2. $\cdots$
- α-β algorithm:
    1. 1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 3$且$|G| = \lceil \frac{n}{3} \rceil$

$\alpha-\beta$算法

Blum等人的算法要求找块中位数集合M的中位数，如果增加算法的表达能力，变为找β分位数，β是根据$\alpha=\frac{k}{n}$调整的值，是否可能可以找到一个策略使得算法比较复杂度为$O(n)$？

算法描述如下

- Blum et al. (1972)的线性算法:
  1.
     1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 3$且$|G| = \lceil \frac{n}{3} \rceil$
     2. 取集合G中每个块的中位数，构成集合M
     3. $m \leftarrow \textsc{Select}(M, \frac{|M|}{2})$
  2. $\cdots$
- α-β algorithm:
  1.
     1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 3$且$|G| = \lceil \frac{n}{3} \rceil$
     2. 取集合G中每个块的中位数，构成集合M

Blum等人的算法要求找块中位数集合M的中位数，如果增加算法的表达能力，变为找β分位数，β是根据$\alpha=\frac{k}{n}$调整的值，是否可能可以找到一个策略使得算法比较复杂度为$O(n)$?
算法描述如下

- Blum et al. (1972)的线性算法:
  1. 1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 3$且$|G|=\lceil \frac{n}{3} \rceil$
     2. 取集合G中每个块的中位数，构成集合M
     3. $m \leftarrow \textsc{Select}(M, \frac{|M|}{2})$
  2. $\cdots$
- α-β algorithm:
  1. 1. 将A划分成若干个块，组成集合G，有$\forall G_1 \in G$，$G_1 \leq 3$且$|G|=\lceil \frac{n}{3} \rceil$
     2. 取集合G中每个块的中位数，构成集合M
     3. $m \leftarrow \textsc{Select}(M, \beta|M|)$
  2. $\cdots$

引理

如果$\alpha=\frac{k}{n}=\frac{1}{2}$，设下一次迭代找第$k'$小数，且$\alpha'=\frac{k'}{n}$，则下一次迭代必然有$\alpha'\leq\frac{1}{4}$或$\alpha'\leq\frac{3}{4}$

$\alpha - \beta$算法

引理

如果$\alpha = \frac{k}{n} = \frac{1}{2}$，设下一次迭代找第$k'$小数，且$\alpha' = \frac{k'}{n}$，则下一次迭代必然有$\alpha' \leq \frac{1}{4}$ 或$\alpha' \leq \frac{3}{4}$

引理表明，如果k靠近中位数，则下一次迭代要找的数将在此次迭代的部分中靠近边缘。

$\alpha-\beta$算法

引理

如果$\alpha=\frac{k}{n}=\frac{1}{2}$，设下一次迭代找第$k'$小数，且$\alpha'=\frac{k'}{n}$，则下一次迭代必然有$\alpha'\le\frac{1}{4}$ 或$\alpha'\le\frac{3}{4}$

引理表明，如果k靠近中位数，则下一次迭代要找的数将在此次迭代的部分中靠近边缘。

设计策略：

- $\alpha<=\frac{1}{4}+\lambda:\beta=\frac{3}{2}\alpha$
- $\alpha\ge\frac{3}{4}-\lambda:\beta=\frac{3}{2}\alpha-\frac{1}{2}$
- $\frac{1}{4}+\lambda<\alpha<\frac{3}{4}-\lambda$: $\beta=\alpha$

其中λ为$\frac{1}{16}$

$\alpha-\beta$算法

引理

如果$\alpha=\frac{k}{n}=\frac{1}{2}$，设下一次迭代找第$k'$小数，且$\alpha'=\frac{k'}{n}$，则下一次迭代必然有$\alpha'\leq\frac{1}{4}$ 或$\alpha'\leq\frac{3}{4}$

引理表明，如果k靠近中位数，则下一次迭代要找的数将在此次迭代的部分中靠近边缘。
设计策略：

- $\alpha <= \frac{1}{4}+\lambda : \beta=\frac{3}{2}\alpha$
- $\alpha \geq \frac{3}{4}-\lambda : \beta=\frac{3}{2}\alpha-\frac{1}{2}$
- $\frac{1}{4}+\lambda<\alpha<\frac{3}{4}-\lambda$: $\beta=\alpha$

其中λ为$\frac{1}{16}$
可以证明，这个策略可以保证找到第k小数的比较复杂度为$O(n)$。

总结和经验

总结：

- 将算法中的常数改为可调参数，然后寻找调整参数的策略，来优化算法
- 手动模拟算法的运行过程，发现规律。

经验：

- 选题很重要，有的题意义不大，很难做而且做出来也发不了文章
- 做研究要有积累和深度，不要四处跳坑。三年以上的时间在一个有意义的问题（可以发顶会文章，如果计划毕业后工作还要注意工业界是否足够关心）上持之以恒地研究，最后的深度会很可观。

Blum, M., Floyd, R. W., Pratt, V., Rivest, R. L., and Tarjan, R. E. (1972). Linear time bounds for median computations. *Stoc'72 Proceedings of the Fourth Annual Acm Symposium on Theory of Computing Acm*, pages 119–124.