طراحی و تحلیل الگوریتم ها

دکتر امیر لکی زاده

استادیار گروه مهندسی کامپیوتر دانشگاه قم

مسئله انتخاب

- Selection Problem

Input: a set of A of n (distinct) number and a number $1 \leq i \leq n$

Output: the element $X \in A$ that is larger than exactly (i-1) other element of A

حالات مختلف: $\begin{cases} i = 1 \rightarrow \text{Min} \\ i = n \rightarrow \text{max} \\ \text{فرد} \; n, \lfloor (n+1)/2 \rfloor \quad \rightarrow \text{Median} \\ \text{زوج} \; n \begin{cases} \lfloor (n+1)/2 \rfloor = i \quad \rightarrow (\text{lower})\text{Median} \\ \lceil (n+1)/2 \rceil = i \quad \rightarrow (\text{upper})\text{Median} \end{cases} \end{cases}$

مسئله انتخاب

- بدست آوردن Min (Max) با n-1 مقایسه ((O(n))
- بدست آوردن Min و Max بطور همزمان :

فرد $n: 3\lfloor n/2 \rfloor$

- عنصر اولی به عنوان Max و Min است سپس به ازای هر جفت، سه مقایسه صورت می گیرد، یکی میان عناصر جفت، دومی میان عنصر کوچکتر جفت با Min و سومی میان عنصر بزرگتر جفت با Max.

زوج $n: 1 + 3\lfloor (n-2)/2 \rfloor$

یک مقایسه روی دو عنصر اول صورت می گیرد و سپس مراحل بعدی مثل حالت فرد =n رخ می دهد.

الگوریتم انتخاب(۱)

```
RANDOMIZED-SELECT(A, p, r, i)
if p = r
    then return A[p]
q ← RANDOMIZED-PARTITION(A, p, r)
k ← q − p + 1
if i = k            ▷ the pivot value is the answer
    then return A[q]
elseif i < k
    then return RANDOMIZED-SELECT(A, p, q − 1, i)
else return RANDOMIZED-SELECT(A, q + 1, r, i − k)
```

- i امین کوچکترین عنصر

الگوریتم انتخاب(۱)

$X_k = \mathrm{I}\{\text{the subarray } A[p\,..\,q] \text{ has exactly } k \text{ elements}\}$,

and so we have

$\mathrm{E}[X_k] = 1/n$.

$$T(n) \le \sum_{k=1}^{n} X_k \cdot (T(\max(k-1, n-k)) + O(n))$$
$$= \sum_{k=1}^{n} X_k \cdot T(\max(k-1, n-k)) + O(n) .$$

$E(x_k)\, E(\max(k-1, n-k)) + an$

فرض $E(T(k)) \le Ck$

$k < n$

$k = 1$	$T(0)$	$T(n-1)$	$T(n-1)$
$k = 2$	$T(1)$	$T(n-2)$	$T(n-2)$
$\vdots$			
$k = n/2$	$T(n/2)$	$T(n/2)$	$T(n/2)$
$\vdots$			
$k = n-1$	$T(n-2)$	$T(1)$	$T(n-2)$
$k = n$	$T(n-1)$	$T(0)$	$T(n-1)$

الگوریتم انتخاب(۱)

$$
\begin{aligned}
&\mathrm{E}[T(n)] \\
&\quad \le \mathrm{E}\left[\sum_{k=1}^{n} X_k \cdot T(\max(k-1, n-k)) + O(n)\right] \\
&\quad = \sum_{k=1}^{n} \mathrm{E}[X_k \cdot T(\max(k-1, n-k))] + O(n) \\
&\quad = \sum_{k=1}^{n} \mathrm{E}[X_k] \cdot \mathrm{E}[T(\max(k-1, n-k))] + O(n) \\
&\quad = \sum_{k=1}^{n} \frac{1}{n} \cdot \mathrm{E}[T(\max(k-1, n-k))] + O(n)
\end{aligned}
$$

الگوریتم انتخاب(۱)

$$\max(k-1, n-k) = \begin{cases} k-1 & \text{if } k > \lceil n/2 \rceil , \\ n-k & \text{if } k \le \lceil n/2 \rceil . \end{cases}$$

$$\mathrm{E}[T(n)] \le \frac{2}{n} \sum_{k=\lfloor n/2 \rfloor}^{n-1} \mathrm{E}[T(k)] + O(n) .$$

الگوریتم انتخاب(۱)

Assume that $T(n) \leq cn$

$$
\begin{aligned}
E[T(n)] &\leq \frac{2}{n}\sum_{k=\lfloor n/2\rfloor}^{n-1} ck + an \\
&= \frac{2c}{n}\left(\sum_{k=1}^{n-1} k - \sum_{k=1}^{\lfloor n/2\rfloor-1} k\right) + an \\
&= \frac{2c}{n}\left(\frac{(n-1)n}{2} - \frac{(\lfloor n/2\rfloor-1)\lfloor n/2\rfloor}{2}\right) + an \\
&\leq \frac{2c}{n}\left(\frac{(n-1)n}{2} - \frac{(n/2-2)(n/2-1)}{2}\right) + an \\
&= \frac{2c}{n}\left(\frac{n^2-n}{2} - \frac{n^2/4-3n/2+2}{2}\right) + an \\
&= \frac{c}{n}\left(\frac{3n^2}{4} + \frac{n}{2} - 2\right) + an \\
&= c\left(\frac{3n}{4} + \frac{1}{2} - \frac{2}{n}\right) + an \\
&\leq \frac{3cn}{4} + \frac{c}{2} + an \\
&= cn - \left(\frac{cn}{4} - \frac{c}{2} - an\right).
\end{aligned}
$$

$cn/4 - c/2 - an \geq 0.$

$n(c/4 - a) \geq c/2.$

$c/4 - a > 0$, i.e., $c > 4a$,

$$n \geq \frac{c/2}{c/4 - a} = \frac{2c}{c - 4a}.$$

الگوریتم انتخاب(۲)

الگوریتم select (i) : O(n)

۱. تقسیم n داده به گروه های ۵ تایی ($\lceil n/5 \rceil$ گروه) $\boldsymbol{O}(\lceil \boldsymbol{n}/\mathbf{5} \rceil)$

۲. مرتب کردن تمامی $\lceil n/5 \rceil$ گروه با مرتب سازی درجی و پیدا کردن Median هر گروه $\boldsymbol{O}(\lceil \mathrm{n}/5 \rceil \times \mathrm{c})$

۳. اجرای بازگشتی الگوریتم برای پیدا کردن Median، Median ها (M^*) $\boldsymbol{T}(\lceil \boldsymbol{n}/\mathbf{5} \rceil)$

۴. اجرای partition در حالتی که M^* ، Pivot می باشد و قرار دادن M^* در مکان K: **T(n)**

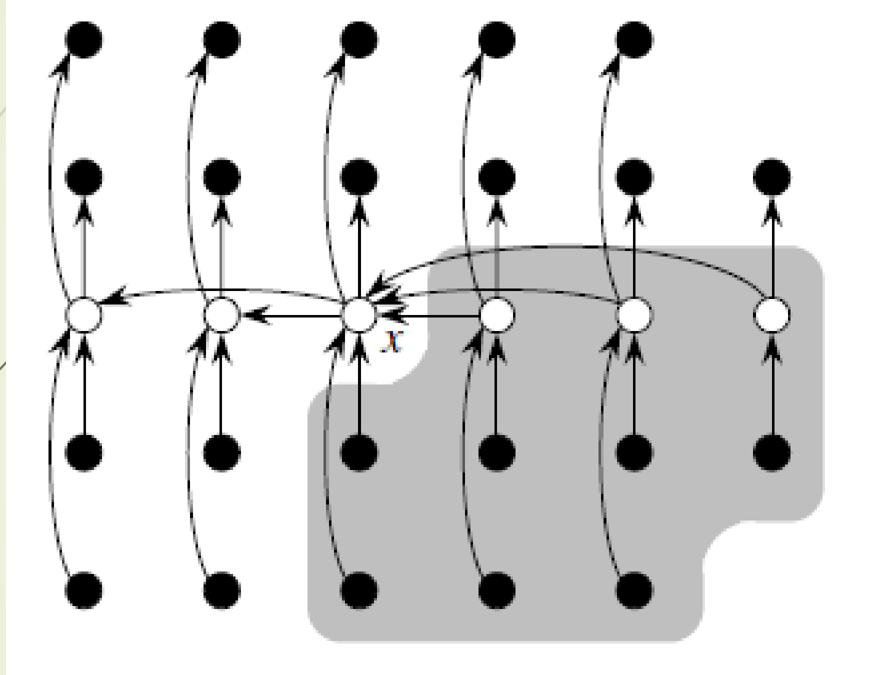
۵. به صورت زیر:

الف) اگر $\mathrm{i} = \mathrm{k}$, M^* جواب مسئله می باشد.

ب) اگر $i < k$، الگوریتم بطور بازگشتی برای انتخاب i امین کوچکترین عنصر در سمت چپ M^* ادامه می یابد.

ج) اگر $i > k$، الگوریتم به طور بازگشتی برای انتخاب $(i - k)$ کوچکترین عنصر در سمت راست M^* ادامه می یابد.

الگوریتم انتخاب(۲)



$$3\left(\left\lceil \frac{1}{2}\left\lceil \frac{n}{5} \right\rceil \right\rceil - 2\right) \geq \frac{3n}{10} - 6 .$$

الگوریتم انتخاب(۲)

$$T(n) \le \begin{cases} \Theta(1) & \text{if } n \le 140 , \\ T(\lceil n/5 \rceil) + T(7n/10 + 6) + O(n) & \text{if } n > 140 . \end{cases}$$

$$\begin{aligned} T(n) &\le c\lceil n/5 \rceil + c(7n/10 + 6) + an \\ &\le cn/5 + c + 7cn/10 + 6c + an \\ &= 9cn/10 + 7c + an \\ &= cn + (-cn/10 + 7c + an) , \end{aligned}$$

which is at most cn if

$$-cn/10 + 7c + an \le 0 .$$

الگوریتم انتخاب

9.3-1

In the algorithm SELECT, the input elements are divided into groups of 5. Will the algorithm work in linear time if they are divided into groups of 7? Argue that SELECT does not run in linear time if groups of 3 are used.

9.3-3

Show how quicksort can be made to run in $O(n \lg n)$ time in the worst case.

9.3-7

Describe an $O(n)$-time algorithm that, given a set S of n distinct numbers and a positive integer $k \leq n$, determines the k numbers in S that are closest to the median of S.

9.3-8

Let $X[1..n]$ and $Y[1..n]$ be two arrays, each containing n numbers already in sorted order. Give an $O(\lg n)$-time algorithm to find the median of all $2n$ elements in arrays X and Y.

مسئله انتخاب

9-2 Weighted median

For n distinct elements $x_1, x_2, \ldots, x_n$ with positive weights $w_1, w_2, \ldots, w_n$ such that $\sum_{i=1}^{n} w_i = 1$, the ***weighted (lower) median*** is the element x_k satisfying

$$\sum_{x_i < x_k} w_i < \frac{1}{2}$$

and

$$\sum_{x_i > x_k} w_i \le \frac{1}{2} .$$

a. Argue that the median of $x_1, x_2, \ldots, x_n$ is the weighted median of the x_i with weights $w_i = 1/n$ for $i = 1, 2, \ldots, n$.

b. Show how to compute the weighted median of n elements in $O(n \lg n)$ worst-case time using sorting.