Selection in expected linear time

- انتخاب i امین کوچکترین عنصر یک آرایه A می‌تواند در $\Theta(n)$ انجام شود.
- تابع RANDOMIZED-SELECT از تابع RANDOMIZED-PARTITION که در Quicksort استفاده کردیم، استفاده می‌کند.
- RANDOMIZED-SELECT متفاوت از Quicksort است چون فقط در یک سمت از partition بازگشت می‌کند

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

بعد از فراخوانی RANDOMIZED-PARTITION آرایه به دو زیرآرایه $A[p..q-1]$ و $A[q+1..r]$ و با pivot مساوی با $A[q]$ تقسیم می‌شود.

- تمام عناصر $A[p..q-1]$ همگی کوچکتر یا مساوی $A[q]$ هستند.
- تمام عناصر $A[q+1..r]$ همگی بزرگتر از $A[q]$ هستند.
- pivot، در واقع k امین کوچکترین عنصر از $A[p..r]$ است که $k = q-p+1$

- اگر عنصر pivot ، i امین کوچکترین عنصر (یعنی $i=k$) است ، $A[q]$ را برمی‌گرداند.
- درغیراینصورت در زیرآرایه‌ای که i امین کوچکترین متعلق به آن است بازگشت می‌کند.

- اگر $i<k$ ، این زیرآرایه $A[p..q-1]$ است و ما می‌خواهیم که i امین کوچکترین عنصر آن را بیابیم.
- اگر $i>k$ ، این زیرآرایه $A[q+1..r]$ است و چون k عنصر در $A[p..r]$ موجود است ، در $A[q+1..r]$ به دنبال $(i-k)$ امین عنصر می‌گردیم.

Analysis :

- worst-case آن $\theta(n^2)$ است ، اگر هر دفعه ناموفق باشیم و هر بار در زیرآرایه‌ای که دقیقاً یک عنصر کوچکتر از زیرآرایه قبلی داشته است جستجو کنیم.

- expected running time

الگوریتم RANDOMIZED - SELECT درحالت average خوب کار می‌کند.
یکی از دلایل آن است که randomized است و ورودی‌های مشخصی نمی‌توانند آن را به حالت worst-case ببرد.

زمان اجرای RANDOMIZED - SELECT یک متغیر تصادفی (random variable) است که با $T(n)$ نشان داده می‌شود ، بنابراین به دنبال یافتن upper bound بر روی مقدار $E[T(n)]$ هستیم.

- RANDOMIZED-PARTITION به طور برابر یکی از عناصر آرایه A را به عنوان pivot برمی‌گرداند.

- برای هر k به صورت $1 \leq k \leq n$، زیرآرایه $A[p..q]$ دارای k تا عنصر است که همه آنها کوچکتر یا مساوی pivot هستند، بنابراین هر عنصر با احتمال $\frac{1}{n}$ می‌تواند انتخاب شود.

- برای $k = 1, 2, \ldots, n$ یک متغیر تصادفی شاخص تعریف می‌کنیم.
(indicator random variable)

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

- احتمال

$$Pr\{\text{subarray } A[p..q] \text{ has exactly } k \text{ elements}\} = \frac{1}{n}$$

$$\Rightarrow \quad E[X_k] = \frac{1}{n}$$

- وقتی که ما RANDOMIZED-SELECT را فراخوانی می‌کنیم. نمی‌دانیم که آیا این الگوریتم فوراً جواب درست را می‌یابد و خاتمه پیدا می‌کند یا روی $A[p..q-1]$ بازگشت می‌زند یا روی $A[q+1..r]$ بازگشت می‌زند. به این بستگی دارد که i امین کوچکترین عنصر کوچکتر یا برابر یا بزرگتر از عنصر pivot $(A[q])$ است یا نه ؟!

- برای بدست آوردن upper bound، فرض می‌کنیم که $T(n)$ به طور یکنواخت افزایشی است
- و i امین کوچکترین عنصر همیشه در subarray بزرگتر است.

- در یک اجرای RANDOMIZED-SELECT ، دقیقا برای یک مقدار $X_k = 1$ از k برقرار است و برای سایر مقادیر $X_k = 0$ است.

- هنگامی که $X_k = 1$ باشد ، اندازه دو زیرآرایه برابر $k-1$ و $n-k$ است.

- برای یک زیرمسئله با سایز n ، RANDOMIZED-PARTITION دقیقا $O(n)$ زمان می‌خواهد.

- بنابراین:

recurrence:

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

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

- امید ریاضی می‌گیریم:

$$E[T(n)] \leqslant E\left[\sum_{k=1}^{n} X_k \cdot T(\max(k-1, n-k)) + O(n)\right]$$

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

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

$$= \sum_{k=1}^{n} \frac{1}{n} E[T(\max(k-1, n-k))] + O(n)$$

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

• اگر n زوج باشد، هر عبارت $T(\lceil n/2 \rceil)$ تا $T(n-1)$ دقیقا دو بار در جمع ظاهر می‌شوند.

• اگر n فرد باشد، این عبارت‌ها دو بار ظاهر می‌شوند و $T(\lfloor n/2 \rfloor)$ یکبار

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

از روش substituation حل می‌کنیم.

○ حدس می‌زنیم $T(n) \leqslant cn$ برای ثابت c که شرایط اولیه هم ارضا می‌کند.

○ فرض می‌کنیم $T(n)=O(1)$ برای n های کوچکتر از یک مقدار ثابت ، درمورد این مقدار ثابت بعداً صحبت می‌کنیم.

○ یک ثابت a برمی‌داریم که تابع توصیف شده‌ی با عبارت $O(n)$ از بالا با an ، برای $n>0$ ، bound شود.

○ با حدس و انتخاب ثابتهای c و a داریم که:

$$E[T(n)] \leqslant \frac{2}{n} \sum_{\lfloor n/2 \rfloor}^{n-1} ck + an$$

$$= \frac{2c}{n} \left(\sum_{k=1}^{n-1} k - \sum_{k=1}^{\lfloor n/2 \rfloor - 1} k \right) + an$$

$$= \frac{2c}{n}\left(\frac{(n-1)n}{2} - \frac{(\lfloor n/2 \rfloor - 1)\lfloor n/2 \rfloor}{2}\right) + an$$

$$\leqslant \frac{2c}{n}\left(\frac{(n-1)n}{2} - \frac{(n/2-2)(n/2-1)}{2}\right) + an$$

$$= c\left(\frac{3n}{4} + \frac{1}{2} - \frac{2}{n}\right) + an$$

$$\leqslant \frac{3cn}{4} + \frac{c}{2} + an$$

$$= cn - \left(\frac{cn}{4} - \frac{c}{2} - an\right)$$

برای تکمیل اثبات مقدار c را اینگونه تعیین می‌کنیم

$$cn/4 - c/2 - an \geqslant 0$$

$$cn/4 - an \geqslant c/2$$

$$n(c/4 - a) \geqslant c/2$$

$$n \geqslant \frac{c/2}{c/4 - a}$$

$$n \geqslant \frac{2c}{c-4a}$$

• فرض کنیم که $T(n) = O(1)$ برای $n < 2c/(c-4a)$ داریم که:

$$E[T(n)] = O(n)$$