جلسه دوم

13:18 Tuesday, September 21, 2021

موضوع: تحلیل بهترین حالت، بدترین حالت و حالت متوسط.

[Average _ worst _ Best] cas Analysis

مسئله ۱) [جستجو]: آرایه A شامل جایگشتی از اعداد ۱ تا n داده شده است.
همچنین عدد x نیز داده شده است.

$1 \le x \le n$

خروجی: اندیس i به گونه‌ای که A[i]=x است.

ورودی: n تا عدد و یک عدد x.

هدف تعیین زمان اجرای الگوریتم برحسب n

راه حل: جستجوی خطی یا sequential search

```
for (i:1 → n)
+ if(A[i]=x)
      return i
```

۰	۱	۲	۳	۴	۵
۵	۴	۹	۷	۲	۱۳

n=۶

x=۷

۱ ← ؟ x=۵

۶ ← x=۱۳

سوال: تعداد مقایسه‌های الگوریتم (خط +) چند تاست؟

Best: ۱ مقایسه. ۱ یعنی $\Omega(1)$ یا

worst case: n مقایسه n یعنی $O(1)$

Average: دقت کنید که برای محاسبه حالت متوسط، نیاز به دانش در رابطه با توزیع ورودی هستیم.

در رابطه با توزیع ورودی هستیم.

مثلا فرض کنید بدانیم که x با احتمال یکسان می‌تواند هرکدام از اعضا باشد.

$$\left.\begin{matrix} \frac{1}{n} \to 1 \\ \frac{1}{n} \to 2 \\ \frac{1}{n} \to 3 \\ \vdots \\ \frac{1}{n} \to n \end{matrix}\right\} \frac{1}{n}\left[1+2+3+\dots+n\right] = \frac{1}{n}\frac{(n)(n+1)}{2} = \frac{n+1}{2} \simeq O(n)$$

مسئله ۲: [جستجوی در آرایه مرتب شده]

ورودی: آرایه مرتب A شامل n عدد و عنصر x $x \in A$

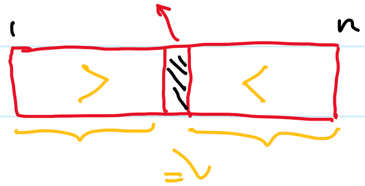
$15=n$ $\boxed{\text{فرض: } 2^k-1=n}$

$31=n$

خروجی: اندیس i به صورتی که $A[i]=x$.

۱- روش جستجوی خطی:

۲- روش جستجوی دودویی: $A[\frac{n}{2}] \overset{?}{=} x$



bin_search(1, n)

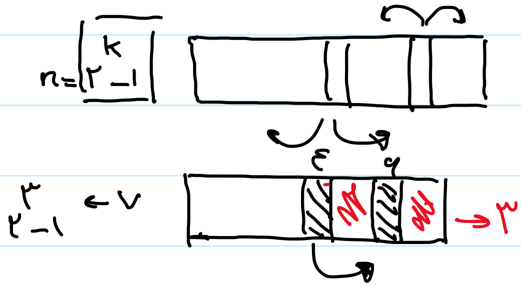
```
bin_search(low, high){
   mid = (low + high)/2
   if(A[mid] > x)
      return(bin_search(low, mid-1))
   if(A[mid] < x)
      return(bin_search(mid+1, high))

   return mid
}
```

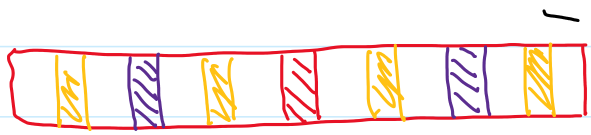
تحلیل

Best case : ۱ مقایسه $\simeq \Omega(1)$

Worst case : $k = \log_2(n+1)$



Average case: فرض : عدد x با احتمال یکسان هرکدام از اعضای آرایه می‌تواند باشد.



$$\frac{1}{n}\times 1 + \frac{2}{n}\times 2 + \frac{4}{n}\times 3 + \frac{8}{n}\times 4 + \dots + \underbrace{\frac{\frac{(n+1)}{2}}{n}}\times \underbrace{\log(n+1)}$$

$$\frac{1}{n}\left[1 + 2\times\boxed{2} + 4\times\boxed{3} + 8\times\boxed{4} + \dots + \frac{n+1}{2}\times\log(n+1)\right] = T$$

برای مقدار T حد پایین و بالا به دست بیاورم.

$$\frac{1}{n}\times\left[\frac{n}{2}\times\log(n+1)\right] \leq T$$

$n = 2^k - 1$

$$\frac{\log(n+1)}{2} \leq T$$

حد بالا :

$$T \leq \frac{1}{n}\left[1\times\log(n+1) + 2\times(\log n+1)) + 4\times(l\dots) + \dots\right]$$

$$T \leq \frac{1}{n}\left[1 \times \log(n+1) + 2 \times (\log n+1) + 4 \times (\log \ldots) + \ldots + \frac{n+1}{2}(\log n+1)\right]$$

$$T \leq \frac{1}{n} \log(n+1) \underbrace{\left[1 + 2 + 4 + \ldots + \frac{n+1}{2}\right]}_{n}$$

$$T \leq \log(n+1)$$

$$= \Theta(\log n)$$

مسئله ۳) مرتب سازی. ورودی: آرایه A شامل n عدد

خروجی: آرایه A به صورت مرتب شده.

* الگوریتم ۱: مرتب سازی حبابی Babble sort

A: ۵ ۶ ۲ ۴ ۱ ←

۵ ۲ ۶ ۴ ۱ الگوریتم حبابی

۵ ۲ ۴ ۶ ۱ سوال: چرا الگوریتم درست است؟

۵ ۲ ۴ ۱ ۶

۲ ۵ ۴ ۱ ۶ * در Pass اول، بزرگترین

۲ ۴ ۵ ۱ ۶ عنصر به انتهای آرایه

۲ ۴ ۱ ۵ ۶ می رود.

⋮ * در Pass دوم، دومین بزرگترین

عنصر به جایگاه n-1 ام می رود.

⋮

← در n-1 Pass آرایه مرتب می شود.

سوال : تعداد مقایسه ها در بهترین ، بدترین و حالت متوسط ؟

فرض : در هر Pass ، اگر swap نداشتیم ← متوقف می‌شویم

بهترین حالت : زمانی که آرایه مرتب باشد : $n-1$ تا مقایسه : $\Omega(n)$

بدترین حالت : زمانی که آرایه نزولی باشد. $n(n-1) = n^2-n$ ← $O(n^2)$

حالت متوسط : هدف : پیدا کردن حد پایین بر روی متوسط تعداد مقایسه‌ها

تعداد swap $\geq$ تعداد مقایسه

* متوسط تعداد swap را پیدا کنیم؟

تعریف ، نابجایی یا inversion : جفت اندیس نا ترتیب به گونه‌ای که $i < j$ و $A[i] > A[j]$.

* تعداد نابجایی ها در هر swap ، 1 واحد کم می‌شود

* آخر کار تعداد نابجایی ها = 0

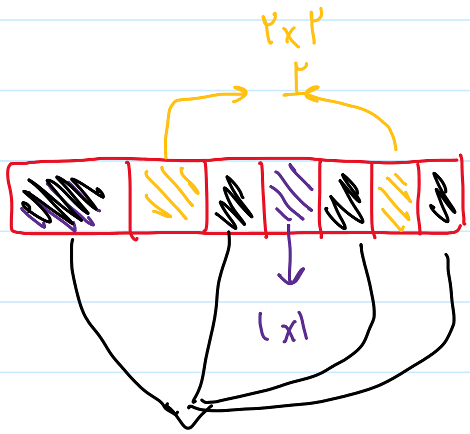
تعداد swap = تعداد نابجایی ها

← متوسط تعداد نابجایی ها ؟

* فرض : همه $n!$ جایگشت مختلف احتمال یکسان دارند.

$n!$ $\binom{n}{2}$

$$\frac{\frac{n!}{2} \times \binom{n}{2}}{n!} = \frac{\binom{n}{2}}{2} = \frac{n(n-1)}{4} = \Omega(n^2)$$



$$n = 7 = 2^{3} - 1$$

$$3 = \log(n+1)$$

$$\frac{7+1}{2} \leftarrow 4 \times 3$$