Subject: جلسه ۴
Year. ۱۴۰۰ Month. ۷ Date. ۷ ( )

موضوع: تحلیل مجانبی و روابط بازگشتی

$f(n) = O(g(n))$: $\exists n_0, c$ $\forall n \geq n_0$ $f(n) \leq c\, g(n)$

c را از یاد گرفتیم که $n_0 = 1$

روابط:

1. $f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$

2. $f_1(n) = O(g_1(n))$ $\quad f_1(n) \times f_2(n) = O(g_1(n) \times g_2(n))$
$f_2(n) = O(g_2(n))$ $\quad f_1(n) + f_2(n) = O(\max[g_1(n), g_2(n)])$

علت $\to$ $x + y \leq 2 \times \max[x, y]$

معمولا O بیشترین استفاده را نسبت به Ω و Θ دارد (چون دارد بدترین حالت را به ما نشان می‌دهد و ما در بررسی الگوریتم‌ها دنبال بدترها هستیم)

$O(1)$ زمان ثابت | $O(\log n)$ زمان لگاریتمی | $O(n)$ زمان خطی | $O(n \log n)$

$O(n^2)$ زمان مربعی | $O(n^3)$ زمان مکعبی | $O(n^k)$ (k ثابت) زمان چند جمله‌ای (Polynomial) | $O(2^n)$ زمان توانی (exponential)

* ساده سازی محاسبه زمان اجرای الگوریتم.

$O(1)$ [$\updownarrow$ $\equiv \to O(1)$

مثال:

for (i = 1 → n)
  ≡] O(1)] $O(n) \times O(1) = O(n)$

$O(n^2 \log n)$
for (i = 1 → n)
  for (j = 1 → n)
    ≡] O(1)
  binSearch (1 ... n)] $O(\log n)$] $O(\log n)$] $O(n \log n)$
seq_search (1, n)] $O(n)$

* $O(n^2 \log n)$

PAPCO

Subject: جلسه ۴
Year. ۱۴۰۰ Month. ۷ Date. ۶

بخش ۲) تحلیل روابط بازگشتی: روش جایگذاری، استقرا، درخت بازگشت، قضیه اصلی که پرکاربردترین

مثال ۱) زمان اجرای کد زیر چقدر است؟ (جست‌وجوی دودویی)

```
bin_search (low, high)
  if (low > high)                    O(1)   T(0)
      return
  mid = (low + high) / 2             O(1)
  if (A[mid] > x)
    bin_search(low, mid-1)           T(n/2)
  if (A[mid] < x)
    bin_search(mid+1, high)
  else
    return(mid)                      O(1)
```

$T(n) = T(\frac{n}{2}) + O(1)$

اندازه آرایه صفر $T(0) = O(1)$

$T(n)$: زمان اجرای الگوریتم در یک آرایه با اندازه n

حل با استفاده از روش جایگذاری:

$T(n) = T(\frac{n}{2}) + O(1) = T(\frac{n}{4}) + O(1) + O(1)$

$= T(\frac{n}{8}) + O(1) + O(1) + O(1) = \dots = T(1) + O(1) + \dots + O(1) = T(0) + O(1) \dots + O(1)$

$= O(1) + O(1) + \dots + O(1) = \underset{?}{O(\log n)} \times O(1) = O(\log n)$

مثال ۲) $T(n) = T(\frac{n}{2}) + O(n)$ $\quad T(1) = O(1)$

$T(n) = T(\frac{n}{2}) + O(n) = T(\frac{n}{4}) + O(\frac{n}{\boxed{2}}) + O(n) = T(\frac{n}{8}) + O(n) + O(n) + O(n) \dots =$

ضریب در O مناسب نیست

$\underbrace{T(1)}_{O(1)} + \underbrace{O(n) + O(n) + \dots + O(n)}_{\log n \times O(n)} = O(n \log n)$

در سوال قبل O(1) بود و در این سوال $O(\frac{n}{2})$ که ما آن مقدار نصف شده در هر مرحله را نادیده می‌گیریم. از دست برود

الگوریتم این O(n) است.

$T(1) = O(1)$ $\qquad$ $T(n) = T(\frac{n}{2}) + O(n)$

$\exists c_1$ داریم: $T(1) \leq c_1$ $\qquad$ $\exists c_2 : T(n) \leq T(\frac{n}{2}) + c_2 n$

$$\begin{aligned}
T(n) &\leq T(\tfrac{n}{2}) + c_2 n \\
&\leq T(\tfrac{n}{4}) + c_2 \tfrac{n}{2} + c_2 n \\
&\leq T(\tfrac{n}{8}) + c_2 \tfrac{n}{4} + c_2 \tfrac{n}{2} + c_2 n \\
&\vdots \\
&\leq T(1) + c_2 \times 2 + c_2 \times 4 + \dots + c_2 \tfrac{n}{2} + c_2 n \\
&\leq c_1 + c_2 \underbrace{[2 + 4 + \dots + \tfrac{n}{2} + n]}_{\leq 2n} \\
&\leq c_1 + c_2 2n \\
&= O(n)
\end{aligned}$$

مثال ۲) مرتب سازی ادغامی یا merge sort.

زیرمسأله (merge): دو آرایه ی A_1 و A_2 به صورت مرتب شده داده شده است. A_1 و A_2 را ادغام کنید.
(فرض $|A_1| + |A_2| = n$)

مثال:

۳	۶	۸	۱۰

۲	۴	۵	۱۱

۲	۳	۴	۵	۶	۸	۱۰	۱۱

زمان الگوریتم ادغام $O(n)$.

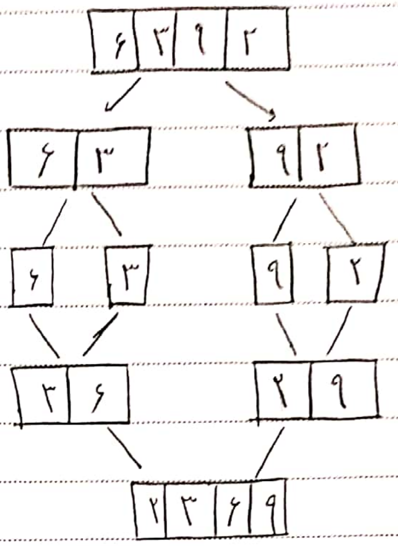
مرتب سازی ادغامی: ۱_ آرایه A را به دو قسمت مساوی تقسیم کنید: A_1, A_2

۲_ A_1 و A_2 را به صورت بازگشتی مرتب کنید

۳_ A_1 و A_2 را ادغام کنید

Subject: جلسه ۴
Year. ۱۴۰۰ Month. ۴ Date. ۶ ( )

```
      merge_sort(low, high)
O(1)      if (low >= high)
              return n
O(1)  mid = (low+high)/2
T(n/2) merge_sort(low, mid)
T(n/2) merge_sort(mid+1, high)
O(n)  merge([low, mid], [mid+1, high])
```



$T(n) = 2T(\frac{n}{2}) + O(n)$ $\qquad T(1) = O(1)$

$T(1) \le c_1$

$T(n) \le 2T(\frac{n}{2}) + c_2 n$

استقرا: نشان می دهیم $T(n) \le c_3 n \log^n$ که در آن $c_3 = c_1 + c_2$. (فرض n توانی از ۲ است)

پایه: $n = 2$

$T(2) \le 2T(1) + c_2(2)$
$\le 2(c_1) + 2c_2 = 2(c_1 + c_2) \times \underbrace{\log_2^2}_{1} = 2c_3 \log_2^2 = c_3 2 \log_2^2$

فرض می کنیم گزاره به ازای همه اعداد کمتر از n صحیح است. برای n داریم:

$T(n) \le 2T(\frac{n}{2}) + c_2 n$
$\le 2 c_3 \frac{n}{2} \log^{\frac{n}{2}} + c_2 n \le (\log^n - 1) c_3 n + c_2 n \le c_3 n \log^n - c_3 n + c_2 n$

$\le c_3 n \log^n = O(n \log^n)$

با استفاده از استقرا می توان نشان داد $T(n) = \Omega(n \log n)$ $\longleftarrow$ $T(n) = \Theta(n \log n)$

PAPCO