

تحلیل پیچیدگی زمانی الگوریتم ها

نام و شماره الگوریتم	پیچیدگی زمانی (حالت معمول)	پیچیدگی زمانی (بدترین حالت)
الگوریتم ۱-۱ جست و جوی ترتیبی		$W(n) = n$
الگوریتم ۱-۲ (جمع کردن عناصر آرایه)	$T(n) = n$	
الگوریتم ۱-۳ مرتب سازی تعویضی	$T(n) = \sum_{i=1}^{n-1} \sum_{j=i+1}^n 1 = \sum_{i=1}^{n-1} n - i$ $T(n) = (n-1) + (n-2) + (n-3) + \dots + 1 = \frac{(n-1)n}{2}$	
الگوریتم ۱-۴ ضرب ماتریس ها	$T(n) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n 1 = \sum_{i=1}^n \sum_{j=1}^n n = \sum_{i=1}^n n^2 = n^3$ $T(n) = n \times n \times n = n^3$	
الگوریتم ۲-۱ جست و جوی دودویی (به روش بازگشتی)		$W(n) = W\left(\frac{n}{2}\right) + 1$ $W(1) = 1$
الگوریتم ۲-۲ مرتب سازی ادغامی		<p style="text-align: center;">پیچیدگی زمانی:</p> $W(n) = W(h) + W(m) + h + m - 1$ <p style="text-align: center;">به ازای $n > 1$ که n توانی از ۲ است:</p> $W(n) = 2W\left(\frac{n}{2}\right) + n - 1$ $W(1) = 0$
الگوریتم ۲-۳ ادغام		$W(h, m) = h + m - 1$
الگوریتم ۲-۴ مرتب سازی سریع	<p style="text-align: right;">به ازای $n > 0$</p> $T(n) = T(0) + T(n-1) + n - 1$ $T(0) = 0$	$W(n) = \frac{n(n-1)}{2} \in \theta(n^2)$
الگوریتم ۲-۷ افراز	$T(n) = n - 1$	
الگوریتم ۲-۸ استراسن	<p style="text-align: right;">به ازای $n > 1$ که n توانی از ۲ است:</p> $T(n) = 7T\left(\frac{n}{2}\right)$ $T(1) = 1$	
تعداد جمع و تفریق های الگوریتم ۲-۸ استراسن		<p style="text-align: right;">به ازای $n > 1$ که n توانی از ۲ است:</p>

	$T(n) = 7T\left(\frac{n}{2}\right) + 18\left(\frac{n}{2}\right)^2$ $T(1) = 1$	
<p>به ازای $n > 5$ که n توانی از ۲ است:</p> $W(n) = 4W\left(\frac{n}{2}\right) + cn$ $W(s) = 0$		<p>الگوریتم ۲-۹</p> <p>ضرب اعداد صحیح بزرگ</p>
<p>به ازای $n > 5$ که n توانی از ۲ است:</p> $3W\left(\frac{n}{2}\right) + cn \leq W(n) \leq 3W\left(\frac{n}{2} + 1\right) + cn$ $W(s) = 0$		<p>الگوریتم ۲-۱۰</p> <p>ضرب اعداد صحیح بزرگ ۲</p>
$T(n) = n \times n \times n = n^3 \in \theta(n^3)$		<p>الگوریتم ۳-۳</p> <p>فلوید برای یافتن کوتاه ترین مسیر</p>
	$\sum_{diagonal=1}^{n-1} [(n - diagonal) \times diagonal] =$ $\frac{n(n-1)(n+1)}{6} \in \theta(n^3)$	<p>الگوریتم ۳-۶</p> <p>حداقل ضرب ها</p>
	$T(n) = \frac{n(n-1)(n+4)}{6} \in \theta(n^3)$	<p>الگوریتم ۳-۹</p> <p>درخت جستجوی دودویی</p> <p>بهبوده</p>
	$T(n) = \sum_{k=1}^{n-2} (n-1-k)k \binom{n-1}{k}$	<p>الگوریتم ۳-۱۱</p> <p>الگوریتم برنامه ریزی پویا برای مسئله فروشنده دوره گرد</p>
	$T(n) = 2(n-1)(n-1) \in \theta(n^2)$	<p>الگوریتم ۴-۱</p> <p>الگوریتم پریم</p>
<p>پیچیدگی زمانی برای مرتب سازی یال ها:</p> $W(m) \in \theta(m \lg m)$		<p>الگوریتم ۴-۲</p> <p>الگوریتم کروسکال</p>
<p>پیچیدگی زمانی برای m بار گذر از حلقه ای که حاوی تعداد ثابتی از فراخوانی روال های equal, find و merge است:</p> $W(m) \in \theta(m \lg m)$		<p>الگوریتم ۴-۲</p> <p>الگوریتم کروسکال</p>
<p>پیچیدگی زمانی برای مقاردهای اولیه:</p> $T(n) \in \theta(n)$		<p>الگوریتم ۴-۲</p> <p>الگوریتم کروسکال</p>
<p>پیچیدگی زمانی در کل:</p> $W(m, n) \in \theta(n^2 \lg n^2) = \theta(n^2 2 \lg n) = \theta(n^2 \lg n)$		<p>الگوریتم ۴-۲</p> <p>الگوریتم کروسکال</p>
		<p>تحلیل الگوریتم ۴-۱</p> <p>پیچیدگی زمانی برای این تحلیل برای همه حالات:</p>

	$T(n) = 2(n-1)^2 \in \theta(n^2)$	برای الگوریتم ۳-۴ الگوریتم دیکسترا
$W(n) \in \theta(n \lg n)$		الگوریتم حریمانه سطح بالا برای روش زمان بندی
$\sum_{i=2}^n [(i-1)+i] = n^2 - 1 \in \theta(n^2)$ در کل: $W(n) \in \theta(n^2)$		الگوریتم ۴-۴ زمان بندی با مهلت معین
$W(n) = \lg n + 1$ اگر n به توانی از ۲ محدود نباشد، در آن صورت: $W(n) = \lfloor \lg n \rfloor + 1 \in \theta(\lg n)$		الگوریتم ۵-۱ جستجوی دودویی

تذکر: پیچیدگی زمانی برای الگوریتم حریمانه سطح بالا در کتاب ذکر نشده که این پیچیدگی برای چه حالتی (معمول، میانگین، بدترین، بهترین) می باشد.

الگوریتم ۱-۲ (جست و جوی دودویی، بازگشتی)

$$\text{حل: } W(n) = \lg n + 1$$

اگر n به توانی از ۲ محدود نباشد، در آن صورت:

$$W(n) = \lfloor \lg n \rfloor + 1 \in \theta(\lg n)$$

الگوریتم ۶-۱ جمله ی n ام فیبوناچی (به صورت بازگشتی)

$$\text{پیچیدگی زمانی: } T(n) > 2^{n/2}$$

الگوریتم ۷-۱ جمله ی n ام فیبوناچی (به روش تکراری)

برای تعیین جمله ی n ام، $(n+1)$ جمله را محاسبه می کند.

اندازه ورودی برای این دو الگوریتم، عبارت از تعداد بیت های لازم برای کد کردن n است که می شود:

$$\lfloor \lg n \rfloor + 1$$

الگوریتم ۲-۲ مرتب سازی ادغامی

به ازای n که توانی از ۲ نباشد:

$$W(n) = W\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + W\left(\left\lceil \frac{n}{2} \right\rceil\right) + n - 1 \Rightarrow W(n) = \theta(n \lg n)$$

بر اساس این قضیه داریم:

$$W(n) \in \theta(n \lg n)$$

تحلیل تعداد ضرب ها در الگوریتم ۸-۲ (استراسن) در حالت معمول

حل این دستور بازگشتی عبارت است از:

$$T(n) = n^{\lg 7} \approx n^{2.81} \in \theta(n^{2.81})$$

تحلیل تعداد جمع و تفریق های الگوریتم ۸-۲ (استراسن) در حالت معمول

حل این دستور بازگشتی عبارت است از:

$$T(n) = 6n^{\lg 7} - 6n^2 \approx 6n^{2.81} - 6n^2 \in \theta(n^{2.81})$$

مقایسه پیچیدگی زمانی الگوریتم استاندارد و الگوریتم استراسن برای n که توانی از ۲ است:

الگوریتم استراسن	الگوریتم استاندارد	
$n^{2.81}$	n^3	ضرب ها
$6n^{2.81} - 6n^2$	$n^3 - n^2$	جمع و تفریق ها

تحلیل پیچیدگی زمانی در بدترین حالت برای الگوریتم ۹-۲ (ضرب اعداد صحیح بزرگ)

$$W(n) \in \theta(n^{\lg 4}) = \theta(n^2) \quad \text{حل:}$$

تحلیل پیچیدگی زمانی در بدترین حالت برای الگوریتم ۱۰-۲ (ضرب اعداد صحیح بزرگ ۲)

$$W(n) \in \theta(n^{\log_2^3}) \approx \theta(n^{1.58}) \quad \text{حل:}$$

تعیین مقادیر آستانه:

دارای دستور بازگشتی زیر می باشد:

به ازای $n > 1$ که n توانی از ۲ است:

$$W(n) = 2W(n/2) + 32n \mu s$$

$$W(1) = 0 \mu s$$

پیچیدگی زمانی الگوریتم ۱-۳ ضریب دوجمله ای با استفاده از روش تقسیم و حل

تعداد جملاتی که الگوریتم برای تعیین $\binom{n}{k}$ محاسبه می کند، برابر است با:

$$2\binom{n}{k} - 1$$

پیچیدگی زمانی الگوریتم ۲-۳ ضریب دوجمله ای با استفاده از برنامه ریزی پویا

$$\frac{k(k+1)}{2} + (n-k+1)(k+1) = \frac{(2n-k+2)(k+1)}{2} \in \theta(nk)$$

پیچیدگی زمانی الگوریتم ۳-۴ دیکسترا

پیچیدگی زمانی برای همه حالات:

$$T(n) = 2(n-1)^2 \in \theta(n^2)$$

توجه: پیچیدگی زمانی برای الگوریتم فلوید و الگوریتم حداقل ضرب ها و الگوریتم درخت جستجوی دودویی بهینه همانطور که در کتاب طراحی الگوریتم ذکر

شده، من هم همانگونه در جدول نوشتم. اما پیچیدگی زمانی برای این سه الگوریتم در جزوه ی که استاد داده، این پیچیدگی زمانی که در جدول نوشتم، استاد در

جزوه ذکر کرده که این پیچیدگی زمانی برای این سه الگوریتم برای تمام حالات می باشد.

پیچیدگی حافظه	پیچیدگی زمانی (بهترین حالت)	پیچیدگی زمانی (حالت میانگین)	نام و شماره الگوریتم
	$B(n) = 1$	$A(n) = \sum_{k=1}^n \left(k \times \frac{1}{n} \right) = \frac{1}{n} \times \sum_{k=1}^n k =$ $\frac{1}{n} \times \frac{n(n+1)}{2} = \frac{n+1}{2}$ $A(n) = \sum_{k=1}^n \left(k \times \frac{p}{n} \right) + n(1-p) =$ $\frac{p}{n} \times \frac{n(n+1)}{2} + n(1-p) =$ $n(1-p) + \frac{p}{2}$	الگوریتم ۱-۱ جست و جوی ترتیبی
		$A(n) = \sum_{p=1}^n \frac{1}{n} [A(p-1) + A(n-p)] + n - 1$	الگوریتم ۲-۶ مرتب سازی سریع
$M(n) = 2 \times n2^{n-1} = n2^n \in \theta(n2^n)$			الگوریتم ۳-۱۱ الگوریتم برنامه ریزی پویا برای مسئله فروشنده دوره گرد
$n(1 + 1/2 + 1/4 + \dots) = 2n$			الگوریتم ۲-۲ مرتب سازی ادغامی
$M(n) = 2 \times n2^{n-1} = n2^n \in \theta(n2^n)$			الگوریتم ۳-۱۱ الگوریتم برنامه ریزی پویا برای مسئله فروشنده دوره گرد