

□ تحلیل (analyze) یک الگوریتم یعنی پیش بینی منابعی که اعدادی الگوریتم به آن نیاز دارد. گاهی منابعی مانند حافظه کامپیوتر، پهنای باند شبکه و یا سخت افزار تأمین می‌شوند. اصلی هستند، ولی در اکثر مواقع این زمان مناسب است که نیاز به پیش بینی دارد.

5. به طور کلی با تحلیل الگوریتم‌های مختلف برای یک مسئله، ما می‌توانیم آن‌ها مشخص می‌شوند.

□ مدل RAM: یک مدل برای تحلیل الگوریتم‌های یک مسئله محاسبه؛ در مدل RAM دستورالعمل‌ها یکی پس از دیگری اجرا می‌شوند و ورودی به دستورات عملیات هستند. تحلیل یک الگوریتم ساده حتی می‌تواند در مدل RAM بسیار دشوار باشد. ابزارهای مورد نیاز می‌تواند شامل ترکیبات، تقریب اعداد، مهارت‌های توانایی تشخیص عبارت‌های تأثیرگذار باشد...

□ تحلیل مرتب سازی درجی: زمان مورد نیاز روشی Insertion-Sort نسبت به ورودی دارد. مرتب سازی هزار عدد بسیار بیشتر از مرتب سازی صد عدد زمان می‌برد. علاوه بر این Insertion Sort ممکن است برای دو ورودی با مقدار کمتر برابر، نسبت به مقدار مرتب بودن آن‌ها در اول کار، به زمان‌های متفاوتی نیاز داشته است.

20 معمولاً زمان مورد نیاز یک الگوریتم با اندازه‌ی ورودی رشد می‌کنند، بنابراین مرتب‌سازی است که زمان اجرای الگوریتم‌ها را به صورت فله‌ای از اندازه‌ی ورودی نشان می‌دهد.

برای اندازه‌ی ورودی طبیعی‌ترین مقیاس مقدار عناصر ورودی است. البته این معیار مساله به مساله باید تعریف شود، برای مثال برای گراف مقدار رئوس و یال‌ها و برای مسائل درختی مساحت، اندازه‌ی ورودی می‌تواند اندازه‌ی ورودی باشد.

مرامیں (Step) اصل اہتمام شدہ۔

زبان اعدای خفا و قتل قضاوت باسد، و بی طرفی می کنیم که اعدای خفا را نام نه زبان⁵

حال صید و صیغ به سرانخ انوسیم Insertion-Sort وانید آید این انوسیم به این حال

۱. پسین ترین زمان صرفه از رویه Insertion-Sort و بهر شکر

15 - وقت کی حقیقتی for و - while - طرز میں

مصطفیٰ فرض بن نسیم (Comments) 1 قاتل - 10 دہشت گرد

INSERTION SORT (A)

خط	کد	تعداد	نوع	تعداد
1	for $j=2$ to $A.length$	1	C_1	n
2	key = $A[j]$	2	C_2	$n-1$
3	// Insert A_j into the Sorted $A_{1..j-1}$	3	X	
4	$i = j-1$	4	C_4	$n-1$
5	while $i > 0$ and $A[i] > key$	5	C_5	$\sum_{j=2}^n t_j$
6	$A[i+1] = A[i]$	6	C_6	$\sum_{j=2}^n t_j - 1$
7	$i = i-1$	7	C_7	$\sum_{j=2}^n t_j - 1$
8	$A[i+1] = key$	8	C_8	$n-1$

زمان اجرای کد الگوریتم برابر است با مجموع زمان اجرای تک تک عبارات عبارتی که به C_i دستورالعمل اصلاح داده و n بار اجرا می شود، در کل زمان اجرای الگوریتم $C_i \times n$ خواهد بود.

سین برای Insertion-Sort: زمان کل که با $T(n)$ نشان داده می شود و n نشان دهنده تعداد ورودی ها است. خواهیم داشت:

$$T(n) = C_1 n + C_2 (n-1) + C_3 (n-1) + C_4 \sum_{j=2}^n t_j + C_6 \sum_{j=2}^n t_j - 1 + C_7 \sum_{j=2}^n t_j - 1 + C_8 (n-1)$$

حالت حال تابع ریاضی برای تابع Insertion-Sort رسیدیم که با تحلیل ویدئو و بررسی آن می توانیم وضعیت الگوریتم مرتب سازی را با عدد ورودی و در پایان ریاضی بررسی کنیم...

برای مثال در Insertion Sort بهترین حالت زمانی اتفاق می افتد که آرایه مرتب شده باشد. برای $n=2, 3, \dots$ و $j=2$ ، درامد های فقط ۵ وقتی که اندازه آرایه $n-1$ را دارد، من بهینه کنیم که $A[j] \leq A[j-1]$. بنابراین برای $n=2, 3, \dots$ و $j=2$ داریم $t=j=2$ و در بهترین حالت زمان اجرا برابر است با

$$T(n) = C_1 n + C_2(n-1) + C_3(n-1) + C_4(n-1) + C_5(n-1) \\ = (C_1 + C_2 + C_3 + C_4 + C_5) n - (C_2 + C_3 + C_4 + C_5)$$

این زمان اجرا را می توان به صورت $a n^2 + b n + c$ نشان داد که در آن a و b ثابت اند و c بستگی به n دارد. بنابراین تابع بالا یک تابع خطی نسبت به n است.

در بهترین حالت زمانی است که آرایه به صورت برعکس مرتب شده باشد یعنی به صورت نزولی یا به محض $A[j] > A[j-1]$ ما یک جابجایی را می بینیم و مرتب شده می شود $A[1 \dots j-1]$ قرار می گیریم و بنابراین برای $n=2, 3, \dots$ و $j=n$ داریم $t=j=n$.

$$\sum_{j=2}^n j = \frac{n(n+1)}{2} - 1 \quad \text{و} \quad \sum_{j=2}^n (j-1) = \frac{n(n-1)}{2}$$

پس در بدترین حالت زمانی اجرای Insertion Sort برابر است با

$$T(n) = C_1 n + C_2(n-1) + C_3(n-1) + C_4 \left(\frac{n(n+1)}{2} - 1 \right) + C_5 \left(\frac{n(n-1)}{2} \right) \\ + C_6 \left(\frac{n(n-1)}{2} \right) + C_7(n-1) \\ = \left(\frac{C_4}{2} + \frac{C_5}{2} + \frac{C_6}{2} \right) n^2 + \left(C_1 + C_2 + C_3 + \frac{C_4}{2} - \frac{C_4}{2} - \frac{C_5}{2} + C_7 \right) n \\ - (C_2 + C_3 + C_4 + C_7)$$

و می توان به ترین زمان اجرا را به صورت $a n^2 + b n + c$ نشان داد که در آن a ، b ، c ثابت های بهینه هستند که به بستری می شود عملیات (C_i) بستگی دارد.

• پس من برای آنکه خواهم آنوردم خود را تحلیل کنم و با مقایسه کنم با سایر آن را وارد
نیای ریاضیات کنم، در نیای ریاضیات اینها را می توانم
به تحلیل من کمک کند.

5. برای اینکار ایدار جدول بنویسید، دفعات تدار و ایدار توابع ریاضی مستقل و وابسته
و ایدار یک تابع $T(n)$ که زمان کن ایدار به حسب مقدار ورودی ایدار کند می تواند تحلیل
را برای ما بسیار ساده کند.

10. پس وقتی می گوییم آنوردمی را تحلیل کنید در واقع هدف ما نوشتن توابع $T(n)$ برای آن
آنوردمی است. و یک تکنیک استفاده از جدول بنویسید دفعات تدار است.

تحلیل بدترین حالت و حالت متوسط

• بدترین حالت در واقع حالتی از حل مساله است که بیشترین زمان صرفه باسد
حداکثر است من از آنوردمی برای صرفه باسد استفاده کنم؟ در واقع آنکه
15 مورد توصیف و تکرار حالت بدترین حالت آنوردمی است و توصیف و تکرار آن به
آن است که بدترین را می بینیم برای اینکار یاد کرده

۱- بدترین حالت زمان ایدار یک آنوردمی، یک کرایه با آن برای زمان ایدار
آنوردمی با هر ورودی است دانستن آن به ما این فواید را میدهد که زمان ایدار
20 هیچ وقت بدتر از آن طول نمی کشد

۱۱- برای بعضی از آنوردمی ها، در اغلب مواقع بدترین حالت ایدار اتفاق می افتد.
۱۲- حالت متوسط زمان ایدار، معمولاً به همان بدترین زمان ایدار در بدترین
حالت است. برای مثال برای یک آرایه و با مساله مرتب سازی در حل آن با مرتب
25 سازی درصده به طور متوسط، نصف زمان $[1, n]$ از $A[n]$ کوکشن و
نصف دیگر از آن به ترتیب است پس به طور متوسط یعنی از زمان آرایه را باید بدترین
کنیم، یعنی $m/2 = n/2$.

عملکردی بیشتر؟

برای تحلیل رویی Insertion - Sort از چند فرض ساده کننده برای ساده شدن تحلیل استفاده شد. ابتدا از فرضیه‌ی واقعی هر محاسبات صرف نظر کرده و آن را با ثابت برای نشان دادیم. سپس دیدیم که حتی این ثابت هم از مقدار n که ما نیاز داریم، جزئیات بیشتری به دست می‌دهند؛ بدترین حالت زمان اجرا برابر است با $an^2 + bn + c$ که در آن a, b, c ثابت‌هایی هستند که به n ها تکیه دارند نه تنها از فرضیه‌ی واقعی که از فرضیه‌ی ساده شده هم چشم پوشی کردیم.

در این جا از یک ساده سازی دیگر نیز استفاده می‌کنیم. این ساده سازی نرخ رشد rate of growth یا مرتبه رشد order of growth زمان اجرا است. و در واقع این سرعت رشد زمان اجرا است که مورد نظر ما است. بنابراین در جدول سرعت رشد فقط جمله‌ی اول (در مثال ما an^2) را در نظر می‌گیریم، چرا که جمله‌ی با درجه پایین در n های بزرگ تأثیر عمیقانی ندارد.

به علاوه در جمله‌ی اول هم از چند ثابت صرف نظر می‌کنیم، زیرا بارش n ، ضرایب ثابت اهمیت خود را در محاسبه‌ی کارایی در حالتی که n از دست می‌دهند.

برای مرتبه سازی در این حقیقت از جمله‌ی با درجات کمتر و ضرایب ثابت صرف نظر کنیم، نقیض عمل n^3 از جمله‌ی اول باقی می‌ماند. می‌گوییم بدترین حالت زمان اجرای الگوریتم مرتبه‌ی n^3 در این صورت است (مخوابیده تکیه‌ای که) است.

معمولاً می‌گوییم الگوریتمی از الگوریتم دیگر کارا تر (سریع تر) است اگر زمان اجرای بدترین حالت آن، سرعت رشد کمتری داشته باشد. برای ورودی‌هایی به اندازه‌ی کافی بزرگ، مثلاً یک الگوریتم از مرتبه‌ی $\theta(n^2)$ در بدترین حالت بسیار سریع تر از یک الگوریتم از مرتبه‌ی زمانی $\theta(n^3)$ اجرا می‌شود.