

فرهاد امان ۶-۹۹۳۱

۱- معیار Θ هنگامی استفاده می شود که می خواهیم کران بالا برای پیچیدگی الگوریتم پیدا کنیم به عنوان مثال می ندانیم بگویم که پیچیدگی زمان الگوریتم مرج سورت از $O(n^3)$ یا از $O(n^2)$ می باشد هر دو گزاره صحیح اند چون کران بالای پیدا کردن کمترین کران بالا برای ما مطلوب است.

۲- برای کران پایین مورد استفاده است و معمولاً برای ما بزرگترین کران

پایین اهمیت دارد.

Θ هنگامی مورد استفاده قرار می گیرد که قصد داریم به طور دقیق پیچیدگی الگوریتم را بدانیم یا می توان گفت که هم کران بالا و هم کران پایین را ارائه دادیم.

به عنوان مثال می گوئیم پیچیدگی مرج سورت از $\Theta(n \log n)$ است.

۲- معیار Θ این معیار برخلاف O که کوچکتر و مساوی بودن پیچیدگی الگوریتم را نشان می داد تنها کوچکتر است و نشان می دهد به عنوان اینکه بگوئیم پیچیدگی مرج سورت از $\Theta(n \log n)$ است گزاره ای نا درست می باشد.

معیار ω برخلاف Θ است و تنها بزرگتر بود پیچیدگی را نشان می دهد.

به عنوان مثال پیچیدگی مرج سورت از $\omega(n)$ است.

در دنیای واقعی گاهی پیش می آید که حافظه دستگاه ما بسیار محدود است مانند برنامه نویسی برای برد ها یا برای موبایل در این مواقع پیچیدگی مکان برای ما اهمیت زیادی دارد اما به طور معمول به علت قیمت ارزان حافظه بیش تمرکز خود را روی کاهش پیچیدگی زمان الگوریتم می گذاریم.

۳ - ابتدا محاسبه می کنیم که کامپیوتر هر عمل جمع را در چند ثانیه می کند.

$$1s = \frac{1}{9 \times 10^{12} + 2 \times 10^6} T(+) \Rightarrow T(+) = \frac{1}{9 \times 10^{12} + 2 \times 10^6}$$

$$T(x) = d \times T(+)$$

$$4n^2 \times T(x) = 20n^2 T(+) = 1s \Rightarrow n \approx 67.82$$

۴ - بهترین روش حرکت خط از ابتدای لیست به انتهای لیست تا زمان که عدد مورد نظر پیدا شود. اگر عدد در ابتدای لیست باشد عدد در $O(1)$ پیدا می شود و اگر در انتها باشد n عمل در لازم است. در حالت کلی میانگین الگوریتم از $O(n)$ عمل می کند.

۵ - از الگوریتم Quick sort استفاده می کنیم ابتدا یک عنصر را به عنوان محور انتخاب کرده پس عناصر را به کوچکتر و بزرگتر از آن تقسیم می کنیم پس انتخاب می کنیم که عنصر مورد نظر ما در کدام سمت قرار گرفته است.

best case = $O(1)$ worst case = $O(n^2)$ average = $O(n)$
 از map استفاده کرده و هر key برابر $value$ آن تعداد تکرار آن است. $O(n)$ پس تعداد یکدست ها داخل آرایه ای رفته و با مرجع سورت $O(n \log n)$ سورت می کنیم پس map مربوط به این آرایه ها برتری گردان $O(n \log n)$

a) True

$$\lim_{n \rightarrow \infty} \frac{n}{\lg n} = \infty$$

- d

b) True

$$\lim_{n \rightarrow \infty} \frac{n \lg n}{n} = \infty$$

c) True

$$\lim_{n \rightarrow \infty} \frac{n^r}{n \lg n} = \infty$$

d) True

$$\lim_{n \rightarrow \infty} \frac{d^{\lg n}}{r^n} = 0$$

a) $T(n) = \Theta(\lg \lg n)$

b) $T(n) = \Theta(\lg \lg \lg n)$

d) $T(n) = \Theta(n)$

- s

فرهاد امان ۶... ۹۹۴۱
۷ - فرض می‌کنیم k یک عدد ثابت باشد.

$$T(n) = k T\left(\frac{n}{k}\right) + O(n)$$

$$a = b^d$$

$$\begin{aligned} a &= k & b &= k \\ d &= 1 \end{aligned}$$

طبق قضیه امله:

$$T(n) \in \Theta(n \lg n)$$

مرتبه‌ی ندیم که در این صورت k تأثیری بر زمان اجرای الگوریتم ندارد.
البته باید زیرماترانه‌ها دارای اندازه یکسان و k عددی ثابت باشد.