

محمد رحیمی ، طراحی الگوریتم ، باهنه سیدآزاد ، ۰۰۱۳۰۰۸۵۹۰۷۰۰۴

برای تعیین زمان اجرای الگوریتم در میان پیچیدگی آن از عباراتی بنام Ω و O استفاده می‌کنیم.

عبار Ω بزرگترین مرتبه در $T(n)$ و مرتبه‌های کوچکتر از آن:

برای مثال (دقیقه‌ای شود زمان اجرای الگوریتم $\Omega(n^2)$ است یعنی الگوریتم هر

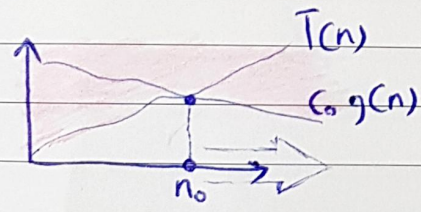
$$T(n) \in \Omega$$

چون اجرای الگوریتم مرتبه اجرایی زمان آن n^2 یا بیش از n^2 است.

عبار Ω بزرگترین مرتبه در $T(n)$ و مرتبه‌های کوچکتر از آن:

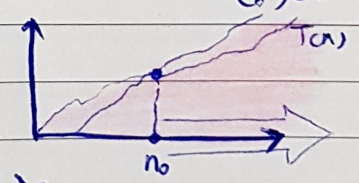
$$T(n) \in \Omega(g(n)) \iff \exists n_0, c > 0 ; \forall n \geq n_0 ; T(n) \leq c \cdot g(n)$$

$$T(n) \in \Omega(g(n)) \iff \lim_{n \rightarrow +\infty} \frac{T(n)}{g(n)} = c \quad \begin{matrix} c > 0 \\ \text{محدودیت} \end{matrix}$$



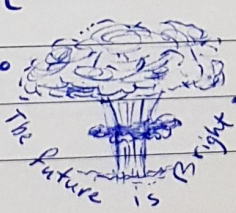
عبار O (بزرگ) شامل بزرگترین مرتبه در $T(n)$ و مرتبه‌های بزرگتر از آن:

$$T(n) \in O$$



$$T(n) \in O(g(n)) \iff \forall n \geq n_0 \exists c ; c > 0$$

$$T(n) \in O(g(n)) \iff \lim_{n \rightarrow +\infty} \frac{T(n)}{g(n)} = 0$$



آینده روشن است

نماد Θ شامل فقط بزرگترین مرتبه در $T(n)$: ✓

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

برای مثال (وقتی گفته می شود زمان اجرای الگوریتم $\Theta(n^2)$ است یعنی الگوریتم ✓

همه جوری اجراء شود مرتبه زمانی اجراء آن فقط (n^2) خواهد بود.

$$T(n) \in \Theta(g(n)) \iff \exists n_0, c_1 > c_2 > 0 \text{ و } \forall n \geq n_0$$

$$\lim_{n \rightarrow \infty} \frac{T(n)}{g(n)} = c$$

$n \rightarrow \infty$

تعریف دیگر:

