Recurrence:

یک recurrence یک تابع است که با ویژگی‌های زیر تعریف می‌شود:

- یک یا تعدادی base case
- خودش با arguments کوچکتر

مثال‌هایی از recurrence:

- $$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n-1)+1 & \text{if } n>1 \end{cases}$$

Solution: $T(n) = n$

- $$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T(\frac{n}{2})+n & \text{if } n\geqslant 1 \end{cases}$$

Solution: $T(n) = n\lg n + n$

- $$T(n) = \begin{cases} 1 & \text{if } n=2 \\ T(\sqrt{n})+1 & \text{if } n>2 \end{cases}$$

Solution: $T(n) = \lg\lg n$

- $$T(n) = \begin{cases} 1 & \text{if } n=1 \\ T(n/3)+T(2n/3)+n & \text{if } n>1 \end{cases}$$

Solution: $T(n) = \Theta(n\lg n)$

پاره‌ای از موضوعات تکنیکی در recurrence:

۱- floors و ceilings ← قابل حذف

۲- تابع exact یا asymptotic

۳- boundary conditions

- در تحلیل الگوریتم‌ها، ما معمولاً هم recurrence و هم solution را بصورت asymptotic notation بیان می‌کنیم به عنوان مثال:

$$T(n) = 2T\left(\frac{n}{2}\right) + \theta(n) \longrightarrow T(n) = \theta(n\lg n)$$

- boundary condition برای n های به قدر کافی کوچک به صورت $T(n)=O(1)$ بیان می‌شود. یعنی می‌گوییم اگر n به قدر کافی کوچک باشد، آن گاه $T(n)=O(1)$ باشد.

- اگر بخواهیم exact solution را به جای asymptotic solution بدست بیاوریم، آنوقت نیازمند آن هستیم که از boundary conditions استفاده کنیم.

- در اینجا و در عمل ما بدنبال asymptotic solution هستیم.

روش‌های قابل قبول برای حل recurrence:

۱- substitution method (روش جایگزینی)

۲- master method

۳- recursion tree (برای محاسبه و حدس و بدست آوردن قابل قبول است)
(به عنوان اثبات قابل قبول نیست)
(به عنوان مقدمه و حدس برای روش substitution method)

روش جایگذینی (substituation method)

گامهای این روش :

۱- حدس زدن Solution

۲- استفاده از induction برای یافتن ثابت‌ها و نشان دادن درستی Solution

مثال:

$$T(n) = \begin{cases} 1 & \text{if } n=1 \\ 2T(\frac{n}{2}) + n & \text{if } n>1 \end{cases}$$

۱- حدس : $T(n) = n\lg n + n$

(در اینجا از تابع exact به جای asymptotic استفاده می‌کنیم، پاسخ هم exact است و نیاز به بررسی base case نیز هست)

۲- Induction (استقرای ریاضی)

پایه استقرا (basis) : $n=1 \Rightarrow n\lg n + n = 1 = T(n)$ ✓

گام استقرا (inductive step) :

فرض استقرا (inductive hypothesis) بیان می‌کند که

$$\text{forall } k<n \quad T(k) = k\lg k + k$$

فرض استقرا را $k=\frac{n}{2}$ می‌گیریم

ثابت می‌کنیم برای $k=n$

$$T(n) = 2T(\frac{n}{2}) + n$$ براساس فرمول

$$T(n) = 2(\frac{n}{2}\lg\frac{n}{2} + \frac{n}{2}) + n$$ براساس فرض

$$= n(\lg n - \lg 2) + n + n$$

$$= n\lg n + n.$$

درحالت general ، از asymptotic notation استفاده می‌کنیم ،

مثلاً می‌نویسیم $T(n) = 2T(\frac{n}{2}) + \theta(n)$

$T(n) = O(1)$ for sufficiently small n

در جواب می‌نویسیم $T(n) = \theta(n \lg n)$

– به شرایط مرزی boundary conditions اهمیت نمی‌دهیم ، base case در اثبات substituation proof نمی‌آید.

Substituation method & asymptotic notation :

– برای ثابت ها ، نام گذاری می‌کنیم.

– حد بالا (O) upper bound ، و حد پایین (Ω) lower bound را جداگانه اثبات می‌کنیم ، اگر نیاز باشد برای ثابت های هرکدام جداگانه نامگذاری می‌کنیم.

مثال: یک recurrence به صورت زیر داده شده است :

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

حدس برای solution برابر عبارت زیر باشد :

$$T(n) = \theta(n \lg n)$$

به دلیل θ ، باید اثبات مشتمل بر دو بخش باشد :

۱- اثبات upper bound یعنی ثابت کنیم که $T(n) = O(n\lg n)$

۲- اثبات lower bound یعنی ثابت کنیم که $T(n) = \Omega(n\lg n)$

① اثبات upper bound :

$$T(n) = 2T(\tfrac{n}{2}) + \theta(n) \Rightarrow T(n) = 2T(\tfrac{n}{2}) + O(n)$$

$\Rightarrow$ for some positive C : $T(n) \leqslant 2T(\frac{n}{2}) + cn$

حدس: طبیعی است که حدس بزنیم

$T(n) \leqslant dn\lg n$

for some positive d (constant)

Substituation:

$$T(n) \leqslant 2T(\tfrac{n}{2}) + cn$$
$$= 2(d\tfrac{n}{2}\lg\tfrac{n}{2}) + cn = dn\lg\tfrac{n}{2} + cn$$
$$= dn\lg n - dn + cn$$
$$\leqslant dn\lg n \qquad \text{if } -dn + cn \leqslant 0$$
$$d \geqslant c.$$

Therefore: $T(n) = O(n\lg n)$

② اثبات lower bound :

for some positive constant c :

$T(n) \geqslant 2T(\frac{n}{2}) + cn$

حدس می‌زنیم : $T(n) \geqslant dn\lg n$ for some positive constant d

substituation:

$$T(n) \geqslant 2T\left(\frac{n}{2}\right) + cn = 2\left(d\frac{n}{2}\lg\frac{n}{2}\right) + cn$$

$$= dn\lg\frac{n}{2} + cn = dn\lg n - dn + cn$$

$$\geqslant dn\lg n \quad \text{if } -dn + cn \geqslant 0$$

$$d \leqslant c$$

Therefore: $T(n) = \Omega(n\lg n)$

$$\Rightarrow \quad T(n) = \Theta(n\lg n)$$

در هنگام استفاده از substituation باید حدس شما exact باشد.

مثلاً $T(n) = 8T\left(\frac{n}{2}\right) + \Theta(n^2)$

برای upper و با حدس $T(n) \leqslant dn^3$ کار نمی‌کند.

باید حدس exact $T(n) \leqslant dn^3 - d'n^2$ اجرا شود.

هر وقت حدس کار نکرد، و فکر می‌کنید که حدس شما درست است یک جمله با order کمتر اضافه کنید.