جلسه اول

13:17 Sunday, September 19, 2021

موضوع: آشنایی با درس ساختمان داده و طراحی الگوریتم ها

محدودیت ها: زمان * - حافظه

تعداد عملیات

مسئله ۱) عدد n داده شده است. مقدار عدد nام دنباله فیبوناچی را به خروجی دهد.

$1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$

$f(n)$: عدد nام فیبو

$$\begin{cases} f(1) = f(2) = 1 \\ f(n) = f(n-1) + f(n-2) \quad \forall n > 2 \end{cases}$$

حل ۱. روش بازگشتی:

شبه‌کد

```
int f(n) {
    if (n <= 2)
+       return 1
    else
+       return f(n-1) + f(n-2)
}
```

* فرض کنید T تعداد دفعاتی است که ۲ خط مشخص شده با علامت + اجرا شود

$T(n)$

$T(1) = 1 = f(1)$

$T(2) = 1 = f(2)$

$T(3) = 1 + T(2) + T(1) = 1 + \overbrace{f(2) + f(1)}^{f(3)} = 3 \geq f(3)$

$T(4) = 1 + T(3) + T(2) \geq 1 + f(3) + f(2) \geq f(4)$

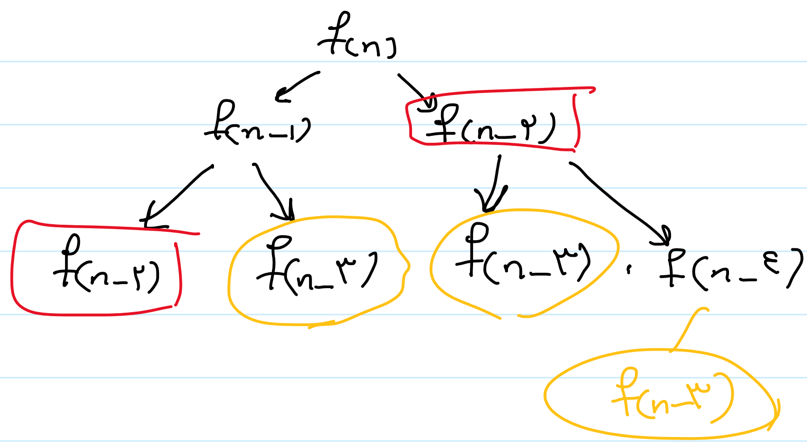
$\vdots$

$T(n) \geq f(n) \simeq \Omega(f(n))$

$T(n) \geq f(n) \simeq \Omega(f(n))$

$f(50) \geq 8 \times 10^9$

$f(51) \longrightarrow$ * رشد نمایی



روش دوم: (برنامه‌نویسی پویا).

در این روش، مقدار $f(i)$ ذخیره می‌شود و از آن برای محاسبه مقدارهای بزرگتر استفاده می‌شود.

f: آرایه با اندازه n

```
+  f[1] = f[2] = 1                 ← 2
   for (i: 3 → n)                  → n-3+1
+      f[i] = f[i-1] + f[i-2]
   return f[n]
```

سوال: خطوطی که با + مشخص شده‌اند در مجموع چند بار اجرا می‌شوند؟

$2 + n - 3 + 1 = 2 + n - 2 = n \simeq O(n)$.

سوال: آیا می‌توان روش دوم را بهینه‌تر کرد؟

از لحاظ حافظه :

```
+ a = b = 1            → ۲
for (i: ۳ → n) {       → ۳(n−۲)
  + temp = b
  + b = a+b
  + a = temp
}
return b
```

$$۳(n-۲) + ۲ = ۳n - ۴ \simeq O(n)$$

سوال : آیا باز هم می‌توان بهتر کرد؟

$b = a+b$ ← $\times$ temp

$a = b - a$

↓

خیلی هم مهم نیست!

راه حل آخر :

قضیه :

$$\boxed{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^n = \begin{bmatrix} f(n+1) & f(n) \\ f(n) & f(n-1) \end{bmatrix}}$$

اثبات : با استفاده از استقرا.

حالت پایه : $n=1$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^1 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} f(2) & f(1) \\ f(1) & f(0) \end{bmatrix}$$

فرض : قضیه برای k برقرار است ، باید ثابت کنیم برای $k+1$ هم برقرار است.

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{k+1} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^{k} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

طبق فرض ↓

$$\begin{bmatrix} f(k+1) & f(k) \\ f(k) & f(k-1) \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$= \underbrace{\begin{bmatrix} f(k+1)+f(k)=f(k+2) & f(k+1) \\ f(k+1) & f(k) \end{bmatrix}}$$

$\square$

* فرض کنید ← $n = 2^k$. در این روش برای محاسبه عدد n ام فیبوناچی، کافی است

* ← k بار ماتریس را در خودش ضرب کنید.

$$\Bigg(\bigg(\Big(\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}^2\Big)^2\bigg)^2 \cdots\Bigg) \quad k$$

$$n = 2^k \rightarrow \begin{bmatrix} 1 & \boxed{1} \\ 1 & 0 \end{bmatrix}^n \quad A$$

$$\underbrace{A^n = (((A^2)^{\overbrace{2})^{\cdots}}^{k\text{ بار}})^2}$$

$$\boxed{n = 2^k}$$

```
int f(n) {
    A = [1 1; 1 0]
    for(i : 1 -> k)
```

مثلاً برای محاسبه عدد 3×10^9 ام فیبوناچی

for(i : 1 → k)

$\rightarrow$ ** $\underbrace{A = A \times A}$ $\rightarrow$ k بار

return $A_{1,2}$

}

مثلا برای محاسبه عدد 3×10^6 ام فیبوناچی

$3 \times 10^9 < 4 \times 10^9 < 4 \times 16^9 < 2^2 \times 2^{36}$

$\boxed{= 2^{38}}$

seddighin.masood@gmail.com