

موضوع: برنامه نویسی پویا ، خردترین سکه و برنامه یزی کارها

در برنامه نویسی پویا مساله به این صورت حل می شود:

- ۱- یک رابطه بازگشتی برای تابع هدف مشخص می شود
- ۲- رابطه بازگشتی ابتدایی زیر مساله ها کوچکتر حل شده و سپس با استفاده از آن ها مساله اصلی حل می شود.

مثال از DS: فلویید وار شال - بهمن فورد، RMQ

مثال ۱) اعداد فیبوناچی ... و ۲۱ و ۱۳ و ۸ و ۵ و ۳ و ۲ و ۱ و ۰

$$\rightarrow f(1) = f(2) = 1$$

$$n > 2 \quad f(n) = f(n-1) + f(n-2)$$

ورودی: عدد n

خروجی: عدد n ام دنباله فیبوناچی

مثال: $n = 5$ خروجی: ۵

روش ۱: بازگشتی //

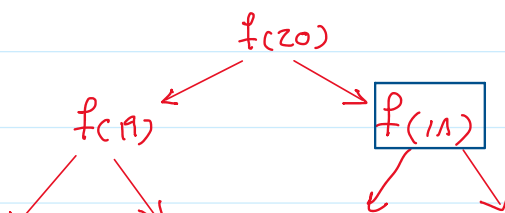
$$f(n) \{$$

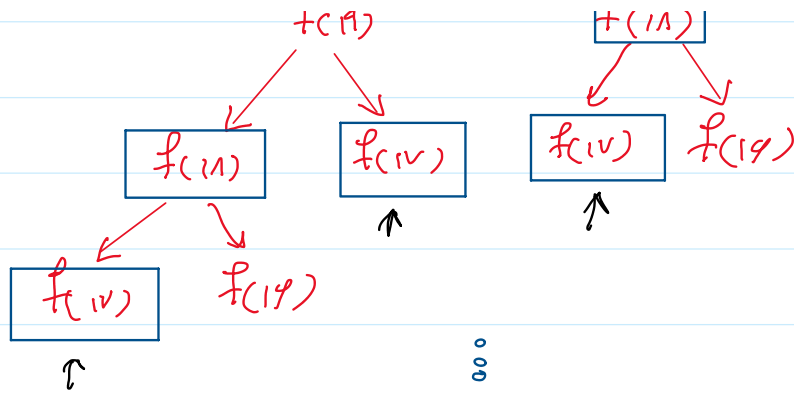
$$if (n \leq 2)$$

$$return 1$$

$$return (f(n-1) + f(n-2))$$

$$\}$$





روش پویا :

$$f[n] = f[n-1] + f[n-2]$$

$$f[1] = f[2] = 1$$

$$\text{for}(i: 3 \rightarrow n)$$

$$f[i] = f[i-1] + f[i-2]$$

$$\} \underline{O(n)}$$

مثال ۱) خرید کردن سکه :

اسکناس ۵ تومانی داده شده است. این اسکناس را توسط سکه‌ها ۱ و ۴ و ۵ تومانی خرید کنید، به طوری که تعداد سکه‌ها کمینه شود.

$$\begin{matrix} ۱۱۱ : ۵ & ۵ & ۵ & ۱۳ \\ ۴ : ۵ & ۴ & ۵ & ۱۳ \end{matrix}$$

حالت کلی : اسکناس ۵ تومانی داده شده است، این اسکناس را توسط سکه‌های c_1, c_2, \dots, c_m و ۵ تومانی خرید کنید به طوری که تعداد سکه‌های استفاده شده کمینه شود.

راه حل پویا : سکه ۱ و ۴ و ۵

$f(i)$: کمترین تعداد سکه مورد نیاز برای خرید کردن i تومان

$$f(100) \left\{ \begin{array}{l} f(99) + 1 \\ f(99) + 1 \\ f(90) + 1 \end{array} \right\} \rightarrow f(100) = \min(f(99), f(99), f(90)) + 1$$

$$\hookrightarrow f(i) = \min(f(i-1), f(i-\epsilon), f(i-\omega)) + 1$$

$$c_m, \dots, c_r, c_1$$

$$f(i) = \min \left(\underbrace{f(i-c_1)}_{\substack{[] \\ i \geq c_1}}, \underbrace{f(i-c_r)}_{\substack{[] \\ i \geq c_r}}, \dots, \underbrace{f(i-c_m)}_{\substack{[] \\ i \geq c_m}} \right) + 1$$

$$f[0] = 0$$

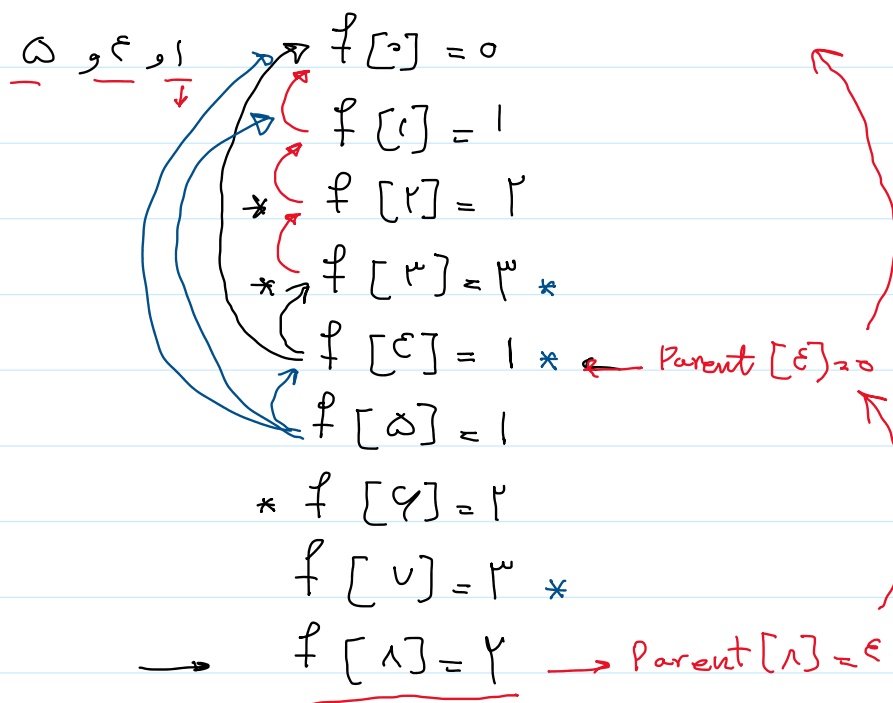
$$\text{for } (i: 1 \rightarrow a)$$

$$f[i] = \infty$$

$$\text{for } (j: 1 \rightarrow m)$$

$$\text{if } (c_j \leq i)$$

$$f[i] = \min(f[i], 1 + f[i - c_j])$$



مثال ۳) برنامه ریزی کارها

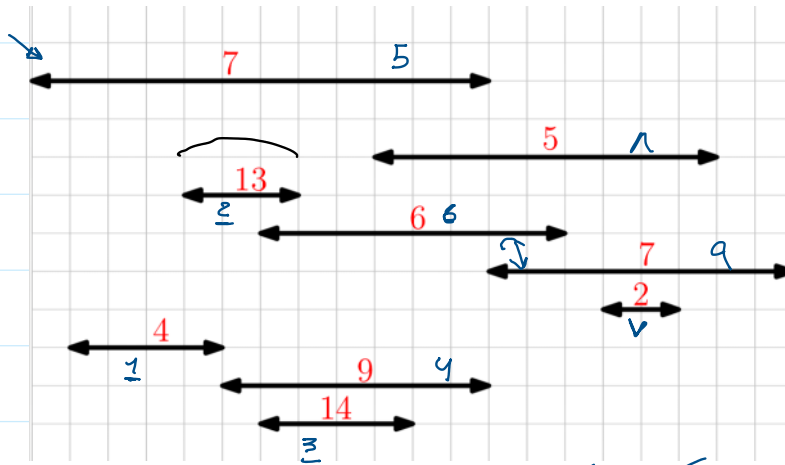
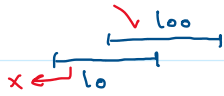
n عدد کار داریم. کارهای زمان شروع s_i و پایان t_i و ارزش v_i است. هدف انتخاب زیر مجموعه S از کارها است به طوری که

۱- تمام کارها مجزا باشند

۲- مجموع ارزش کارهای انتخاب شده بیشینه شود.

راه حل حریصانه برای مساله: در هر مرحله کاری را انتخاب کنیم که میزان زنگ آگن

کمینه باشد و با کارهای قبلی تداخل نداشته باشد.



$$P[9] = 5$$

$$P[8] = 2$$

$$P[7] = 6$$

کاری را انتخاب کنیم که چگالی بیشینه داشته باشد.



جواب بجهت؟

فرض: کارها بر حسب انتها مرتب شده هستند. \leftarrow sort

$f(i)$: جواب بیشینه برای تا کار اول

$$f(9) = \max(f(5) + 7, f(8))$$

بازه 9 انتخاب شده، بازه 8 انتخاب نشده

به ازای i ها، $P[i]$: کار با اندیس بیشینه که قبل از شروع کار تا تمام می شود

هزینه برای $P[i]$: کار با اندیس پیشینه که قبل از شروع کار نامهای می شود

$$f(i) = \begin{cases} v_i + f(p[i]) & \text{شامل بازه نام هست} \\ f(i-1) & \text{شامل بازه نام نیست} \end{cases}$$

$$\begin{cases} f[0] = 0 \\ \text{for } (i: 1 \rightarrow n) \\ f[i] = \max(v_i + \underbrace{f[p[i]]}, f[i-1]) \\ \text{return } \underline{f[n]} \end{cases} \quad O(n)$$

* سوال پایانی: $P[i]$ ها را در چه زمانی می توان حساب کرد؟
کل $P[i]$ ها را می توان در $O(n)$ حساب کرد.
