1 Rekurencie

este raz riesenia niektorych rekurencii z cvik. mame danu rekurenciu napr $T(n) = aT(\frac{n}{b}) + k$. idea postupu je postupne rozpisovat cleny $T(\frac{n}{b})$ .... teda $T(\frac{n}{b}) = aT(\frac{n}{b^2}) + 1$ ... dokym v tom neuvidime nejaky tvar riesenia.

dolezite je sa zamerat na to, kolko krat to takto mozeme prepisat (inymi slovami, kolko vnoreni bude mat dana rekurzia).

Theorem 1.1.

$$T(n) = \begin{cases} T\left(\frac{n}{2}\right) + 1 & \textit{if } \; n > 1 \\ 1 & \textit{if } \; n = 1 \end{cases}$$

Proof. niekolko prvych iteracii

$$T(n) = T\left(\frac{n}{2}\right) + 1 = \left(T\left(\frac{n}{2^2}\right) + 1\right) + 1 = \left(\left(T\left(\frac{n}{2^3}\right) + 1\right) + 1\right) + 1 =$$
$$= T\left(\frac{n}{2^3}\right) + 1 + 1 + 1$$

atd. jedine co robime je, ze za $T(\frac{n}{2^i})$ dosadime $T(\frac{n}{2^{i+1}}) + 1$ presne podla zadania rekurencie.

dokedy mozeme tento krok opakovat? zrejme dovtedy, az kym nedosiahneme pri iterovani $T(1)$. vsimnite si, premenna n v $T(n)$ sa po kazdom kroku zmensi na polovicu.

takze po i iteraciach mame $T(n) = T(\frac{n}{2^i})$ a plus nejake jednotky (konkretne i jednotiek, ale k tomu neskor). a teda spomenuta podmienka na zastavenie iteracie je, ze

$$\frac{n}{2^i} = 1 \qquad \Rightarrow \qquad i = \log_2 n$$

v $\log_2 n$-tej iteracii narazime na $T(1)$.

OK, dalej si vsimnime, ze kazda iteracia vyrobi jednu jednotku (najlepsie to uvidite tak, ked si sami niekolko iteracii spravite). no a s tym, ze iteracii mozeme spravit len $\log_2 n$, tychto jednotiek tam bude $\log_2 n$.

$$T(n) = T(1) + \log_2 n = O(\log n)$$

$\square$

Theorem 1.2.

$$T(n) = \begin{cases} 2T\left(\frac{n}{2}\right) + 5 & \textit{if } \; n > 1 \\ 5 & \textit{if } \; n = 1 \end{cases}$$

Proof. opat rozpiseme niekolko clenov

$$T(n) = 2T\left(\frac{n}{2}\right) + 5 = 2\left(2T\left(\frac{n}{2^2}\right) + 5\right) + 5 = 2\left(2\left(2T\left(\frac{n}{2^3}\right) + 5\right) + 5\right) + 5$$

skute si rozpisat este niekolko dalsich clenov.

ako v predoslom priklade, aj tu sa premenna n v $T(n)$ po kazdej iteracii zmensi na polovicu. z rovnakeho dovodu je preto pocet moznych iteracii zas $\log_2 n$.

teraz nam ta ale zavadzia ta dvojka v $T(n) = \mathbf{2}T(\frac{n}{2})+5$, ktora tam predtym nebola. opat ale vidno, ze po i-tej iteracii to bude vyzerat takto

$$\begin{aligned} T(n) &= 2^i T\left(\frac{n}{2^i}\right) + 5.2^{i-1} + 5.2^{i-2} + ... + 5.2 + 5 = \\ &= 2^i T\left(\frac{n}{2^i}\right) + 5\left(2^{i-1} + 2^{i-2} + ... + 2 + 1\right) \end{aligned}$$

vznikol nam tam geometricky rad, ktoreho sucet lahko zistime podla nasledujuceho vztahu

$$1 + q + q^2 + q^3 + ... + q^M = \frac{q^{M+1} - 1}{q - 1}$$

v nasom pripade je $q = 2$

no a ked spravime tych iteracii $\log_2 n$, potom dostaneme

$$T(n) = 2^{\log_2 n} T(1) + 5\left(2^{(\log_2 n)-1} + 2^{(\log_2 n)-2} + ... + 2 + 1\right) = 2^{\log_2 n} T(1) + 5\frac{2^{log_2 n} - 1}{2 - 1}$$

dalsou dolezitou logaritmicko-exponencialnou identitou je tato

$$a^{\log_a n} = n$$

pouzijuc tento vztah dostavame

$$T(n) = nT(1) + 5\frac{n-1}{2-1} = O(n).$$

□

Theorem 1.3. *mozeme mat aj nieco taketo*

$$T(n) = \begin{cases} 3T(\frac{n}{5}) + 2 & \textit{if} \ \ n > 1 \\ 5 & \textit{if} \ \ n = 1 \end{cases}$$

Proof. najpr sa pozrime, kolko iteracnych krokov treba spravit. $T(n)$ sa meni na $T(\frac{n}{5})$, teda n sa zmeni na patinu

$$n \to \frac{n}{5} \to \frac{n}{5^2} \to \frac{n}{5^3} \ldots$$

a teda zastavime sa ked bude platit

$$\frac{n}{5^i} = 1$$

co je splnene pre $i = \log_5 n$ ... oprati predoslemu prikladu sa zmenil zaklad logaritmu.

dalej sa pozrime ci nam tam pri tych iteraciach vznikne. no za kazdou iteraciu sa k $T(\frac{n}{5^i})$ pribali jedna 3ka (spravte si ich niekolko nech to naozaj vidite).

no a potom tam zostane takyto sucet (vseobecne pre i-tu iteraciu)

$$\begin{aligned} T(n) &= 3^i T\left(\frac{n}{5^i}\right) + 2.3^{i-1} + 2.3^{i-2} + \ldots + 2.3 + 2 = \\ &= 3^i T\left(\frac{n}{5^i}\right) + 2\left(3^{i-1} + 3^{i-2} + \ldots + 3 + 1\right) \end{aligned}$$

posledna iteracia (pre $i = \log_5 n$) vyzera takto

$$T(n) = 3^{\log_5 n} T(1) + 2\left(3^{(\log_5 n)-1} + 3^{(\log_5 n)-2} + \ldots + 3 + 1\right)$$

este sa "'zbavime"' tej geometrickej postupnosti. podla vzorca z predosleho prikladu (kvocient q je teraz 3) ju mozeme prepisat ako

$$T(n) = 3^{\log_5 n} T(1) + 2\frac{3^{\log_5 n} - 1}{3 - 1} = 3^{\log_5 n} T(1) + 3^{\log_5 n} - 1$$

vsimnite si, ze teraz sme dostali vyraz

$$3^{\log_5 n}$$

nemozeme pouzit predosli postup kedze cisla 3 a 5 su rozne. mame ale dalsiu log identitu (vsimnite si ake su zaklady tychto logaritmov)

$$\log_a c = \frac{\log_b c}{\log_b a}$$

takze $\log_5 n$ prepiseme na

$$\log_5 n = \frac{\log_3 n}{\log_3 5}$$

a spojenim tychto veci mame

$$3^{\log_5 n} = 3^{\frac{\log_3 n}{\log_3 5}} = \left(3^{\log_3 n}\right)^{\frac{1}{\log_3 5}} = n^{\frac{1}{\log_3 5}}$$

ked sa vratime spat k nasej rekurencii, uz vieme pokracovat

$$T(n) = 3^{\log_5 n} T(1) + 3^{\log_5 n} - 1 = n^{\frac{1}{\log_3 5}} T(1) + n^{\frac{1}{\log_3 5}} - 1 = O(n^{\frac{1}{\log_3 5}})$$

pre porovnanie, $\log_3 5 \approx 1,5$ a $\frac{1}{\log_3 5} \approx 0,66$ takze tato zlozitost je mensia ako $O(n)$ z predosleho prikladu. □

Theorem 1.4. *znama a velmi casta*

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

Proof. tato je v skutocnosti velmi lahka. treba si len uvedomit, ze pri iterovani sa teraz bude menit aj to n mimo T ... teda

$$T(n) = 2T\left(\frac{n}{2}\right) + n = 2\left(2T\left(\frac{n}{2^2}\right) + \frac{n}{2}\right) + n = ...$$

spravte si este dve, potom si to roznasobte a uz budete vidiet, ako to pokracuje. potom si staci iba uvedomit, kolko iteracii budete mat a je to.

mne to vyslo $T(n) = O(n \log n)$. □

oplati sa s tymto trochu pohrat teraz ako potom na pisomke. az dostanete nejaky cit pre to, ako sa tieto veci spravaju, pozrite si v skriptach *Master metodu*, ktora umoznuju pohodlne riesenie niektorych typov rekurencii. inu verziu tejto metody mozete najst v materialoch na stranke.

na zaver jeden priklad z minuleho midtermu. mame takuto funkciu

```
c=array[1..N]

function(a,b)
          ret=0

          for i=a to b do
                    ret=ret+c[i]
                    ret=ret mod 2

          if |b-a|>8 then do
                    for i=0 to 3 do
                              ret=ret + f(a + i(b-a)/6, a + (i+3)(b-a)/6)

          return ret mod 2
```

zaujima nas zlozitost volania $f(1, N)$.

Proof. nasa funkcia pracuje s polom c, pricom na vstupe dostane nejaky interval $< a, b >$. oznacme cislom n dlzku tohoto intervalu, teda $n = b - a$.

dalej si vsimnime, ze funkcia najpr nieco pocita ... prvy cyklus ma dlzku n ... a potom sa v druhom cykle rekurzivne vola. kedze toto rekurzivne volanie je v cykle, bude tychto volani viac ... presne 4.

pozrime sa teraz na tieto rek. volania:

$$\begin{aligned}
i = 0 \quad &\Rightarrow \quad f(a, a + 3\frac{b-a}{6}) \\
i = 1 \quad &\Rightarrow \quad f(a + \frac{b-a}{6}, a + 4\frac{b-a}{6}) \\
i = 2 \quad &\Rightarrow \quad f(a + 2\frac{b-a}{6}, a + 5\frac{b-a}{6}) \\
i = 3 \quad &\Rightarrow \quad f(a + 3\frac{b-a}{6}, a + 6\frac{b-a}{6})
\end{aligned}$$

nas konkretne zaujima, ake dlhe intervaly sa pouziju pre jendotlive volania (pripominame, ze dlzka behu prvej casti tela funkcie je umerna dlzke vstupneho intervalu, a preto pocet krokov vypoctu $f(A, B)$ mozeme urcit pomoc dlzky vstupneho intervalu $< A, B >$). lahko vidno, ze zakazdym je dlzka intervalu $\frac{b-a}{2} = \frac{n}{2}$.

OK, staci to uz len dat dokopy. mame rekurentnu funkciu pre ktoru plati, ze ak na vstupe dostala interval dlzky N, potom spravi radovo N krokov a 4 krat sa zavola pre intervaly dlzky $\frac{N}{2}$. podla predoslych prikladov teda mame

$$T(N) = 4T\left(\frac{N}{2}\right) + bN$$

kde b je nejaka konstanta.

skuste si najpr sami spocitat zlozitost. potom si vysledok overte podla Master metody v skriptach. $\square$