

# Predpočítanie

Alebo rozprávanie o tom,  
ako sa prihotoviť  
na nepriateľa

MišoF., 2006



# O čom to bude?

Pamätáte si film **Sám doma**?

- chlapec sám doma
- prichádzajú zloději
- pripraví sa na nich
- *v konštantnom čase* vie spustiť pascu



# Príprava arzenálu

---

Začneme tým, že si nazbierame nejaké užitočné veci

(10 metrov odizolovaného drôtu,  
bejzbolku, švihadlo, spinkovačku,  
jedovato vyzerajúceho pavúka  
a za vrečko gumených macíkov?)

# Binary search(?)

```
int binary_search(vector<int> pole, int hladam) {
    int zac = 0;           // prvý prvok
    int kon = pole.size()-1; // posledný prvok
    while (zac != kon) {
        int stred = (zac+kon)/2;
        if (pole[stred] < hladam) zac = stred;
        else kon = stred;
    }
    if (pole[zac]==hladam) return zac;
    return -1; // nie je tam
}
```

# Binary search(?)

```
int binary_search(vector<int> pole, int hladam) {
    int zac = 0;           // prvý prvok
    int kon = pole.size()-1; // posledný prvok
    while (zac != kon) {
        int stred = (zac+kon)/2;
        if (pole[stred] < hladam) zac = stred;
        else kon = stred;
    }
    if (pole[zac]==hladam) return zac;
    return -1; // nie je tam
}
```

Čo ak hľadáme číslo 10 v poli

{1, 3, 5, 10, 12}?

$[zac, kon] : [0, 4] \rightarrow [2, 4] \rightarrow [2, 3] \rightarrow [2, 3] \rightarrow \dots$

# Half-open intervals

**Problém:**  $\pm 1$  chyby

**Trik:** polo-otvorené intervaly

Interval tvaru  $\langle a, b \rangle = \{x \mid a \leq x < b\}$ .

Ak popisujeme úsek poľa, udávame:

- index ukazujúci **na** začiatok
- index ukazujúci **tesne za** koniec

napríklad celé STL používa takýto prístup

# Binary search!

Hľadáme v intervale  $\langle zac, kon \rangle$ .

**Invariant:**  $pole[zac] \leq hľadám < pole[kon]$

```
int binary_search(vector<int> pole, int hľadám) {
    int zac = 0;           // prvý prvok
    int kon = pole.size(); // ZA posledný prvok

    if (pole[zac] > hľadám) return -1;

    while (kon-zac != 1) {
        int stred = (zac+kon)/2;
        if (A[stred] < hľadám) zac = stred;
        else kon = stred;
    }
    if (A[zac]==hľadám) return zac;
    return -1; // nie je tam
}
```

# Prefixové sumy

**Problém:** dané je  $N$ -prvkové pole, bude treba vedieť rýchlo povedať k intervalu súčet prvkov v ňom

index:	0	1	2	3	4	5
A:	3	6	1	4	2	3

**Riešenie:**

Pre každé  $k$  (od 0 do  $N$ ) spočítame súčet prvých  $k$  prvkov poľa.



# Prefixové sumy

Pre každé  $k$  (od 0 do  $N$ ) spočítame súčet prvých  $k$  prvkov poľa.

index:	0	1	2	3	4	5	
A:	3	6	1	4	2	3	
S:	0	3	9	10	14	16	19

$S[k]$  je súčet poľa  $A$  pre interval  $\langle 0, k \rangle$ .

Ako nájsť súčet pre všeobecný interval  $\langle a, b \rangle$ ?

# Prefixové sumy

Súčet v intervale  $\langle 2, 5 \rangle$ :

index:	0	1	2	3	4	5
S[5]:	3	6	1	4	2	3
S[2]:	3	6	1	4	2	3
rozdiel:	3	6	1	4	2	3

**Záver:**

Súčet prvkov poľa  $A$  v intervale  $\langle a, b \rangle$   
je rovný  $S[b] - S[a]$ .

# Prefixové sumy 2D

**Problém:** dané je 2D pole rozmerov  $R \times S$ , bude treba vedieť rýchlo povedať k obdĺžniku súčet prvkov v ňom

A:	0	1	2	3	4	5
0	5	7	2	1	4	6
1	5	2	1	5	7	0
2	0	2	4	5	1	1
3	3	5	6	7	8	2
4	5	5	2	1	1	1



# Prefixové sumy 2D

---

## Riešenia:

- Hrubá sila v  $O(RS)$ .
- Prefix sumy v riadkoch,  $O(R)$ .
- Ide aj lepšie?

# Prefixové sumy 2D

Toto chceme:

**[0,0]**



**[a,b]**

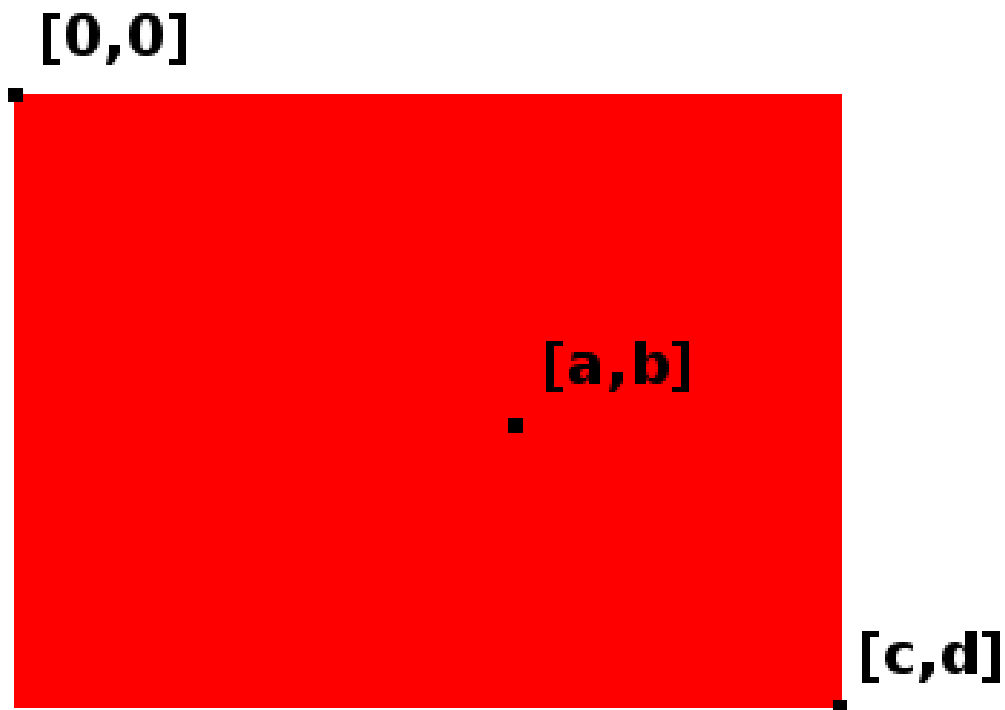


**[c,d]**



# Prefixové sumy 2D

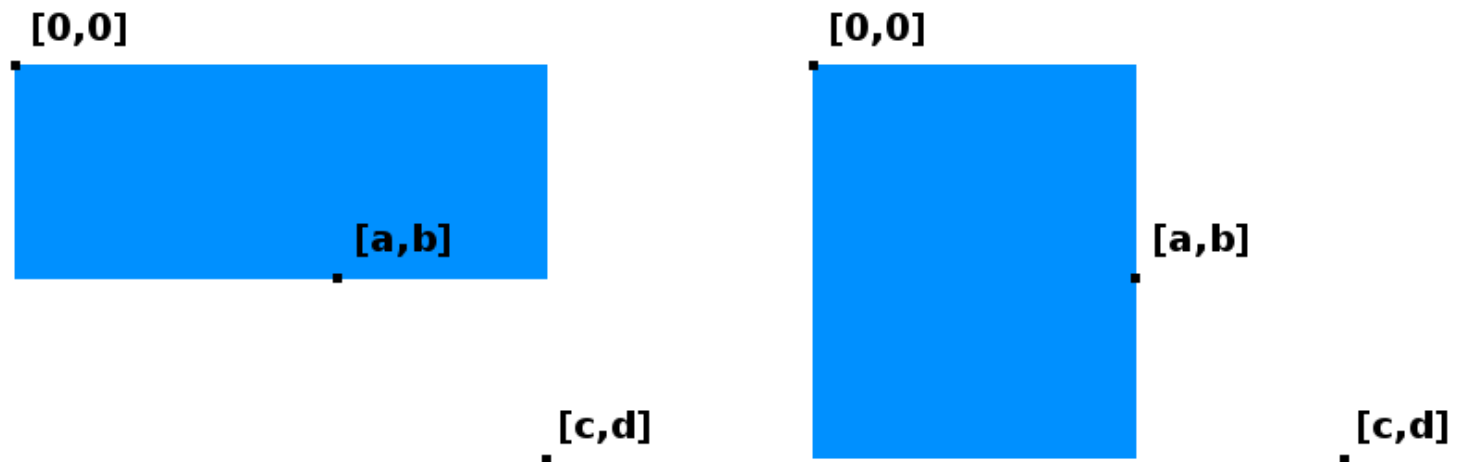
Takéto všetky nech vieme:



Pomôže nám to?

# Prefixové sumy 2D

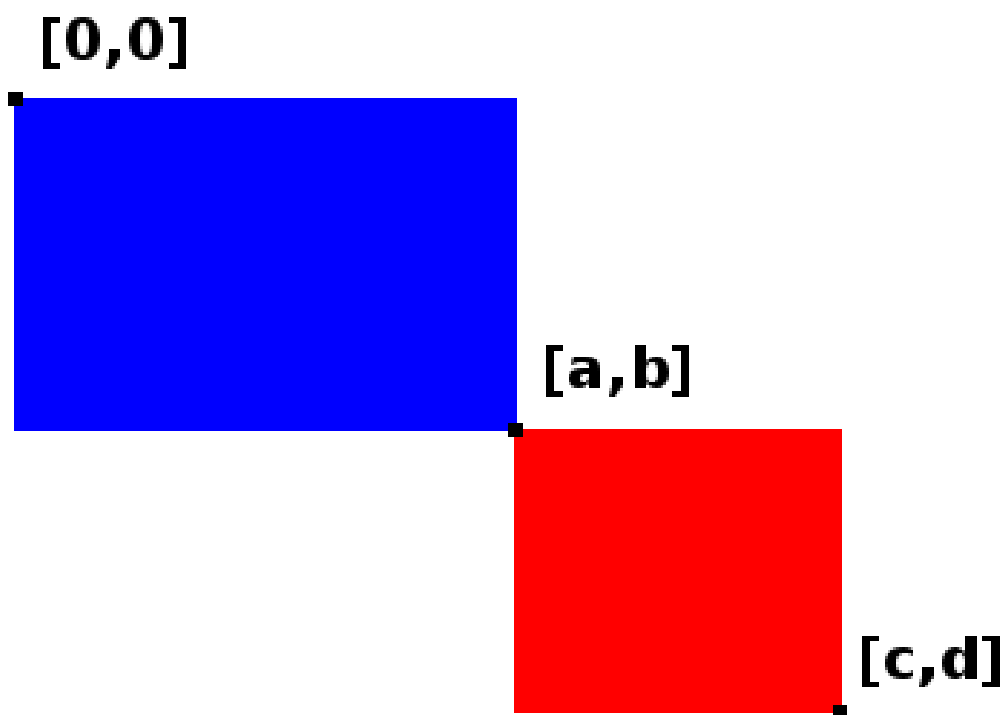
Od minulého obrázku odčítame tieto:



Máme už to, čo sme chceli?

# Prefixové sumy 2D

Takmer dobre. Zostalo toto:



Už len pripočítame ten ľavý horný kus a je to.



# Prefixové sumy 2D

S poľami to vyzerá takto:

Nech  $S[r][s]$  je súčet na políčkach v riadkoch  $\langle 0, r \rangle$  a stĺpcoch  $\langle 0, s \rangle$ .

Súčet v riadkoch  $\langle r_1, r_2 \rangle$  a stĺpcoch  $\langle s_1, s_2 \rangle$  dostaneme takto:

$$S[r_2][s_2] - S[r_1][s_2] - S[r_2][s_1] + S[r_1][s_1]$$

Posledná otázka: odkiaľ vziať  $S$ ?

# Prefixové sumy 2D

Odkiaľ vziať  $S$ ?

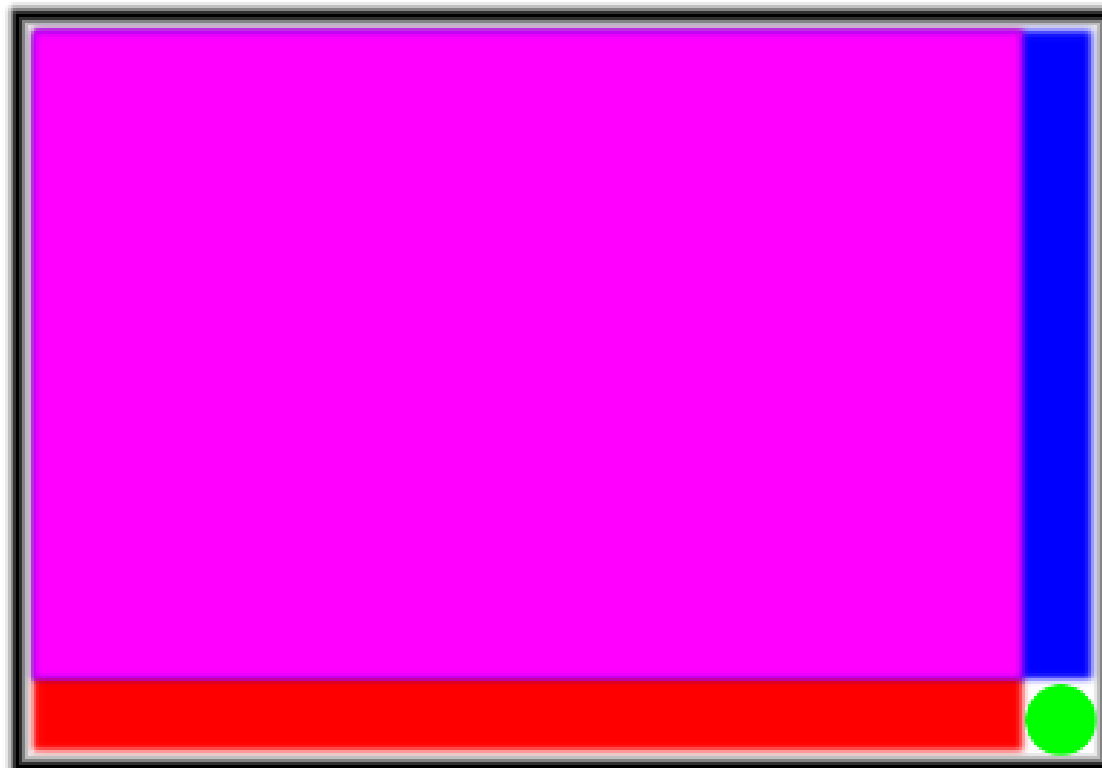
Pole  $S$  vypočítame presne tak isto!



# Prefixové sumy 2D

Odkiaľ vziať  $S$ ?

Pole  $S$  vypočítame presne tak isto!



# Prefixové sumy 2D

Odkiaľ vziať  $S$ ?

Pole  $S$  vypočítame presne tak isto!

Pre "súčet" jedného políčka platí:

$$A[r - 1][s - 1] = \\ S[r][s] - S[r - 1][s_2] - S[r][s - 1] + S[r - 1][s - 1]$$

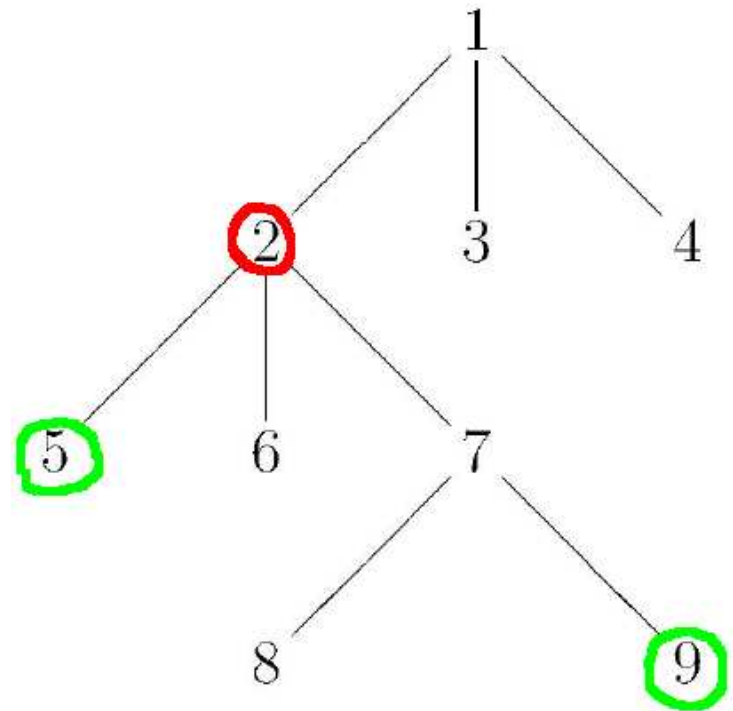
A toto upravíme na:

$$S[r][s] = \\ A[r - 1][s - 1] + S[r - 1][s_2] + S[r][s - 1] \\ - S[r - 1][s - 1]$$

# Predok v strome

## Problém:

daný je zakorenený strom  
bude treba vedieť rýchlo povedať  
k dvom vrcholom  
ich najbližšieho spoločného predka



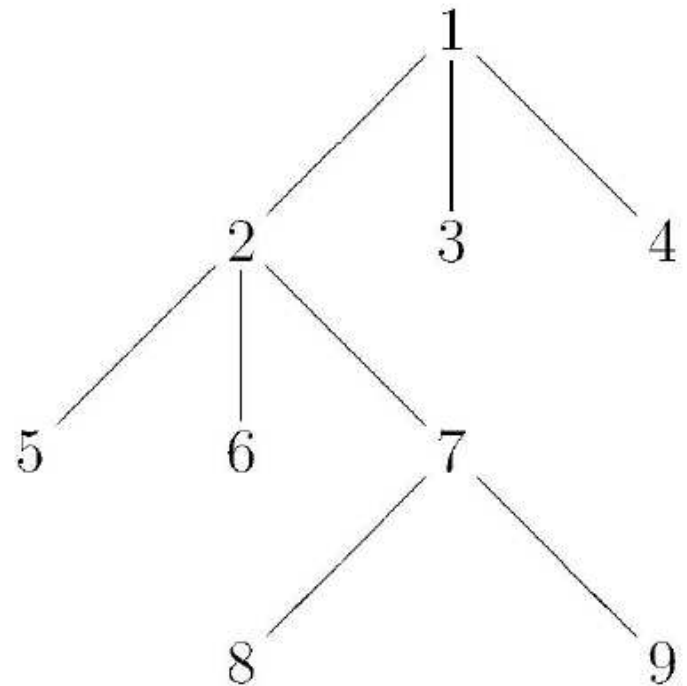
Pre vstup 5, 9 je výstup 2.

# Predok v strome

## Triviálne riešenie:

lezieme z oboch dohora  
farbíme cestu za sebou  
keď niekto príde na ofarbené,  
vyhrali sme

Čas:  $O(N)$ .

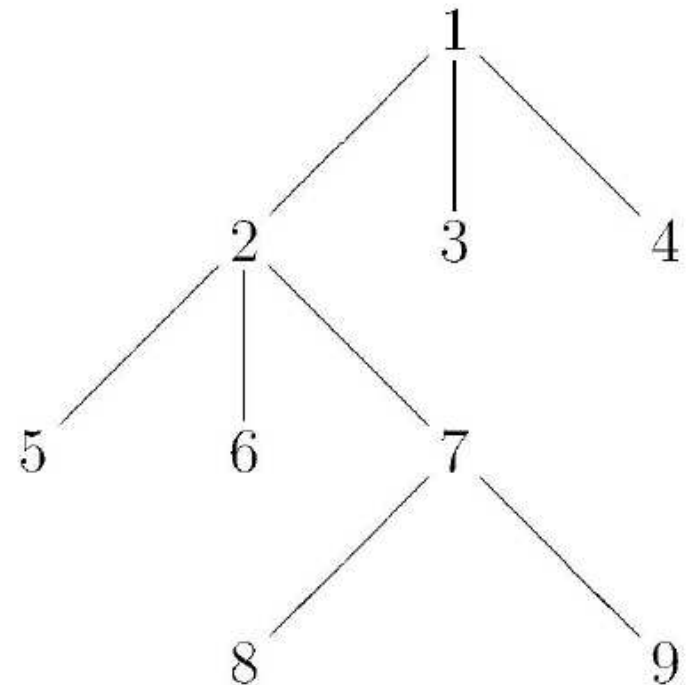


# Predok v strome

## Zrýchlenie:

chceme íe z' rýchlejšie  
predrátame:

Kam sa dostanem z tohoto vrchola  
 $2^k$  krokmi dohora?



Na začiatku vieme pre každý vrchol len  $k = 0$ .

Keď vieme odpovede pre nejaké  $k$ , pre  $k + 1$  spočítame v  $O(N)$ :

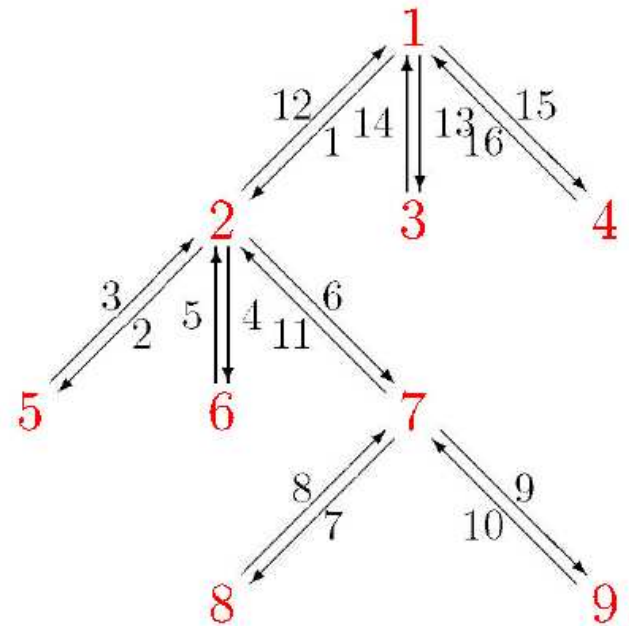
$$kam(v, 2^{k+1} \text{ krokov}) = kam(kam(v, 2^k \text{ krokov}), 2^k \text{ krokov})$$

# Predok v strome

## Lepšie riešenie:

na začiatok

prehľadáme strom do hĺbky:



krok:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
vrchol:	1	2	5	2	6	2	7	8	7	9	7	2	1	3	1	4	1
hĺbka:	0	1	2	1	2	1	2	3	2	3	2	1	0	1	0	1	0

## Prvá návšteva vrcholu:

vrchol:	1	2	3	4	5	6	7	8	9
krok:	0	1	13	15	2	4	6	7	9



# Predok v strome

Nech  $P[v]$  je čas  
prvej návštevy vrcholu  $v$

Najbližší predok  $a$  a  $b$

=

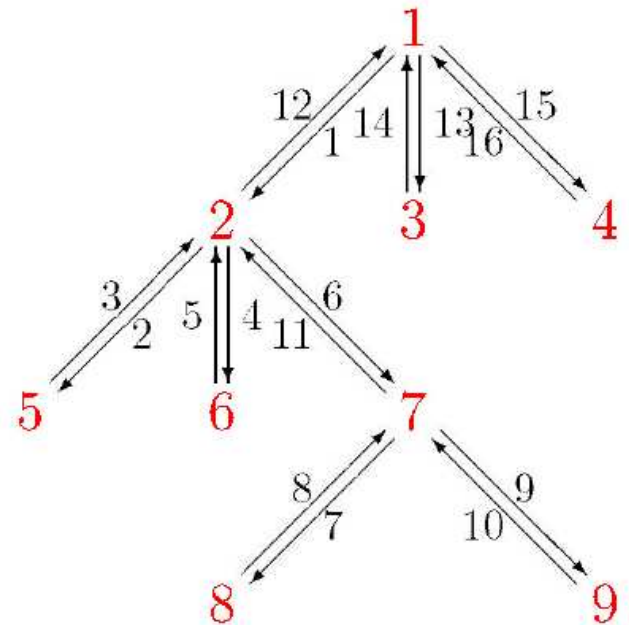
koreň podstromu kde sú

=

"najplytší" vrchol

navštívený medzi  $P[a]$  a  $P[b]$ .

Ako ho nájsť?



# Predok v strome

Záznam z DFS:

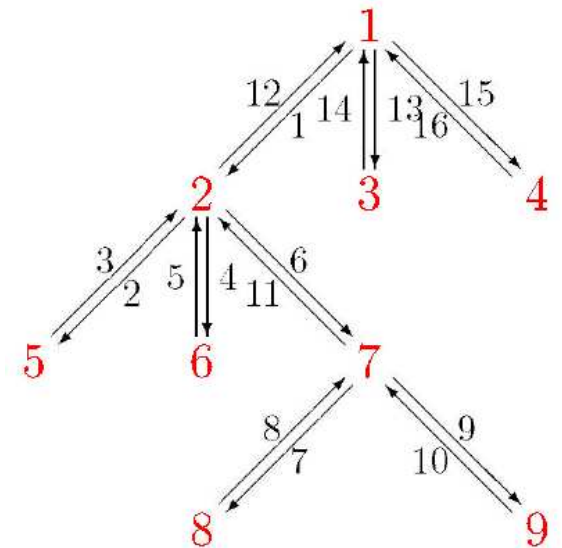
krok:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
vrchol:	1	2	5	2	6	2	7	8	7	9	7	2	1	3	1	4	1
hĺbka:	0	1	2	1	2	1	2	3	2	3	2	1	0	1	0	1	0

Prvá návšteva vrcholu:

vrchol:	1	2	3	4	5	6	7	8	9
krok:	0	1	13	15	2	4	6	7	9

Chceme:

nájsť minimum na úseku daného poľa



# Predok v strome

**Máme:** pole

hĺbka: 0 1 2 1 2 1 2 3 2 3 2 1 0 1 0 1 0

**Chceme:** pre ľubovoľný úsek vedieť,  
kde je v ňom (jedno) minimum.

**Predrátame:**

pre každý začiatok  $a$

pre každú dĺžku  $2^k$

kde je minimum v  $\langle a, a + 2^k \rangle$

**Časová zložitosť:**

predrátanie v  $O(N \log N)$

# Predok v strome

**Vieme:**

kde je minimum v  $\langle a, a + 2^k \rangle$  (pre každé  $a$  a  $k$ )

**Chceme:**

kde je minimum v  $\langle a, b \rangle$  (pre každé  $a$  a  $b$ )

**Finta:**

najväčšie  $k$  také, že  $2^k \leq b - a$

minimum z  $\langle a, b \rangle$  = menšie z:

minimum z  $\langle a, a + 2^k \rangle$  a minimum z  $\langle b - 2^k, b \rangle$

**Príklad:**

0	1	2	1	2	1	2	3	2	3	2	1	0	1	0	1	0
0	1	2	1	2	1	2	3	2	3	2	1	0	1	0	1	0
0	1	2	1	2	1	2	3	2	3	2	1	0	1	0	1	0



# Sufixové pole

---

**Máme:** dlhý text

**Chceme:**

- rýchlo vyhľadávať
- nájsť opakujúci sa úsek
- . . . .



# Suffixové pole

---

**Finta:** *suffixy* obsahujú všetko info o texte

**Slovo:** matematika má suffixy:

matematika

atematika

tematika

ematika

matika

atika

tika

ika

ka

a

# Sufixové pole

## Utriedime sufixy:

1. a
2. atematika
3. atika
4. ematika
5. ika
6. ka
7. matematika
8. matika
9. tematika
10. tika

0123456789

matematika

## Sufixové pole:

zač.: 0 1 2 3 4 5 6 7 8 9

por.: 7 2 9 4 8 3 10 5 6 1

alebo ekvivalentne

## poradie sufixov:

9 1 5 3 7 8 0 4 2 6



# Sufixové pole

Ako ho vyrobiť?

V  $i$ -tom kroku utriedime podľa prvých  $2^i$  znakov.

Jeden krok v  $O(N) \Rightarrow$  celé v  $O(N \log N)$ .

Prvý krok:

$$\text{poradie}(x) = \text{ASCII}(\text{retazec}[x])$$





# Sufixové pole

$(n + 1)$ . krok:

Už máme utriedené podľa prvých  $2^n$  znakov.

Sufix začínajúci  $x$ -tým písmenom  
je na pozícii  $poradie(x)$ .

Majme dva sufixy  $x, y$ ,  
kde  $poradie(x) = poradie(y)$ .

Chceme porovnať ich **d'alších**  $2^n$  znakov.



# Sufixové pole

---

$(n + 1)$ . krok na príklade:

1. a
2. atika
2. atematika
3. ematika
4. ika
5. ka
6. matematika
6. matika
7. tematika
8. tika

# Sufixové pole

$(n + 1)$ . krok na príklade:

1. a
2. atika
2. atematika
3. ematika
4. ika
5. ka
6. matematika
6. matika
7. tematika
8. tika

Treba porovnať:

2. atematika
2. atika

Ale veď už máme porovnané:

3. ematika
4. ika



# Sufixové pole

$(n + 1)$ . **krok** všeobecne:

Sufix začínajúci písmenom  $x$

dostane kľúč  $[poradie(x), poradie(x + 2^n)]$

Utriedime sufixy podľa kľúča

Nanovo očísľujeme (nové *poradie*).



# Obed

Dvanásť hodín, čas obeda