

Felkel@fel.cvut.cz

www.cgg.cvut.cz/~felkel/DSA

<http://pitboul.wz.cz>

arithmetic series

$$s = (a_1 + a_n) \cdot \frac{n}{2}$$

$$a_{i+1} = d + a_i$$

$$a_n = d \cdot (n-1) + a_1$$

sum of first n natural numbers?

$$\begin{bmatrix} 1 & 2 & 3 & 4 & \dots & N \\ 2 & 3 & 4 & 5 & \dots & N+1 \\ \vdots & & & & & \vdots \\ N & & & & & 2N-1 \end{bmatrix}$$

$$s = 1 + \dots + N = \frac{n^2 + n}{2}$$

$$s_1 = \frac{(n+1) \cdot n}{2} = \frac{n^2 + n}{2}$$

$$s_2 = \frac{(n+3) \cdot n}{2} = \frac{n^2 + 3n}{2}$$

$$\vdots$$
$$s_m = \frac{(2n-1+n) \cdot n}{2} = \frac{3n^2 - n}{2}$$

$$\Sigma = \left(\frac{n^2 + n}{2} + \frac{3n^2 - n}{2} \right) \cdot \frac{n}{2}$$

for (i=0; i<100; i++)

for (j=0; j<50; j++)

sum += j

100x
50x
5000x

very much

for (i=0; i<80; i++)

for (j=i; j<90; j++)

sum += j

3240x

- pole přeseďte nulové nakonec bez změny pořadí

$0 \times \times 000 \times \times \times 0 \rightarrow 00000 \mid \times \times \times \times \times$

pročítat od konce, (určuje i jestli 0 už byla) ^{reklóg} pokud
je dál na x - přehodit s nulou a uharovat ho
na nulu posunout doleva

bylaNula = false;

for (int i = p.length - 1; i > -1; i--) {

if (bylaNula) {

if (p[i] != 0) {

p[misto] = p[i];

p[i] = 0;

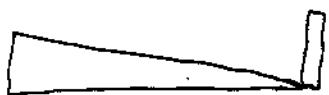
misto--;

}

else

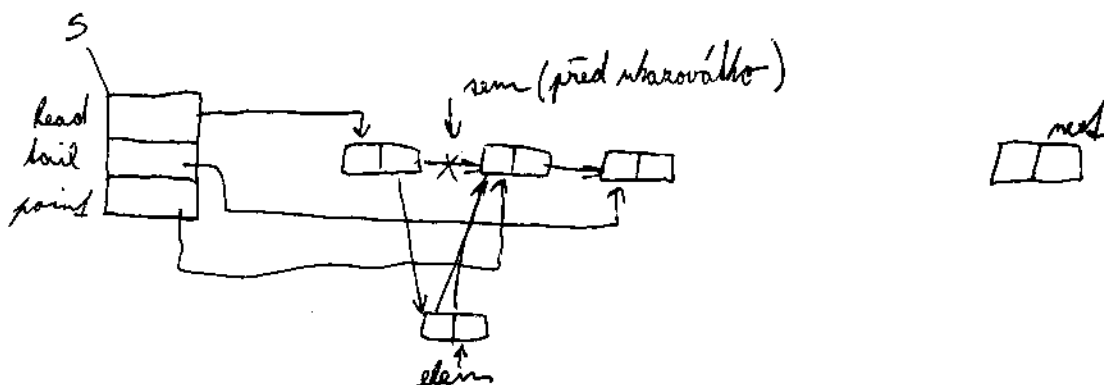
{ if (p[i] == 0) { bylaNula = true; misto = i; }

pole N různých prvků



složnost insert-sout?
necelupne

- insert do seznamu



vytváří se nový prvek, do něj se
překopíruje obsah prvku na point do nového,
ten vloží se na prvek point a do prvku
na point vloží ^{novou} hodnotu

relativní případy

poslední prvek - mění se tail
point je null - seznam prázdný nebo ukazuje
na konec?

EAS*Y*QUE***ST***IO*N***

hárobník

příměno PUSH(P)

* TOP, POP

hárobník EASYQUESTYON

SYEUGRTSAONIE

EASY

a) EASY : E*A*S*Y*

b) YSAE : EASY****

c) ASYE : EA*S*Y**

d) AYES = nejde

pravidelný binární strom

počet listů $\frac{n+1}{2}$ 2. počet uzlů

TreeSort (pole *p)

{ tree t;

for (i=0; i < length(p); i++)

insert (t, p[i])

inorder (t) → p

$$\left. \begin{array}{l} n \\ \log n \\ n \end{array} \right\} \begin{array}{l} n^2 \\ n \log n \end{array}$$

$O(n^2)$ nevyužívá. BVS

$O(n \log n)$ využívá. BVS

$$f(x, y) = \begin{cases} 0 & \text{pokud } x=0 \vee y=0 \\ f(x-1, y) + f(x, y-1) + f(x-1, y-1) + 1 & \text{jinak} \end{cases}$$

Hodnoty přepočítat do pole a pak vrátit výsledek

		x					
		0	1	2			
y ↓	0	0	0	0	0	0	...
	1	0	1	2	3	4	5
	2	0	2	6	12	20	30
		0	3	12	31	64	115
		0	4	20	64		
		0	5	30	115		

```

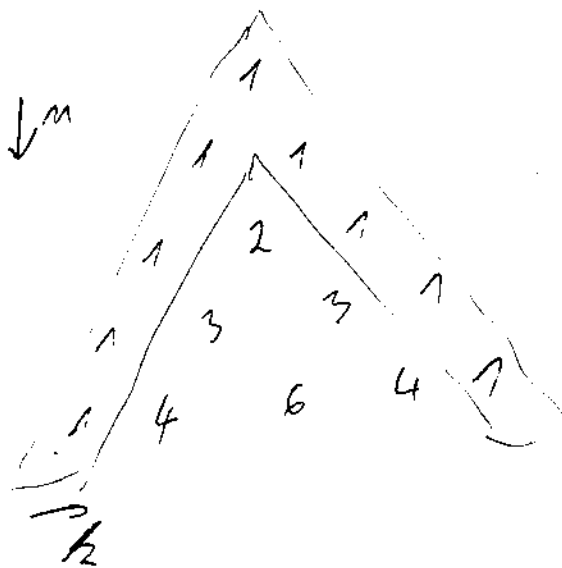
int rekurePolom() {
    int [][] pole = new int [x+1][y+1];

    for (int i=0; i<=x; i++) pole[i][0]=0;
        i      i      y      0      j

    for (int i=1; i<=x; i++) {
        for (int j=1; j<=y; j++) {
            pole[i][j] = pole pole[i-1][j] + pole[i][j-1] + pole[i-1][j-1];
        }
    }
    return pole[x][y]
}

```

Pascalio Δ



$\text{Bin}(n, k) = \text{Bin}(n-1, k) + \text{Bin}(n-1, k-1)$ / 1 pro $n=0$ v $k=0$

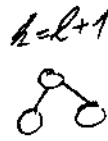
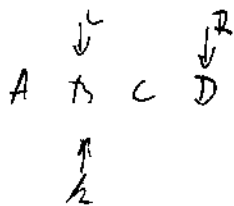
holkrāt volāna pri $\text{Bin}(6, 4)$

Optimální strom

n malých optimálních subtřešeni spočítat
optimální řešení

- A 0,1
- B 0,15
- C 0,2
- D 0,3
- E 0,1
- F 0,1
- G 0,05

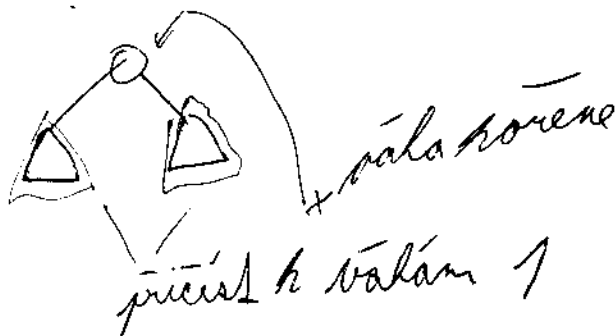
		$\xrightarrow{R}$						
		A	B	C	D	E	F	G
A	0	(A)	(B)					
B	0	0,1	0,35					
C	0	0	0,15	(C)				
D	0	0	0	0,2	(D)			
E	0	0	0	0	0,3	(E)		
F	0	0	0	0	0	0,1	(F)	
G	0	0	0	0	0	0	0,1	(G)



?

$$L \leq R$$

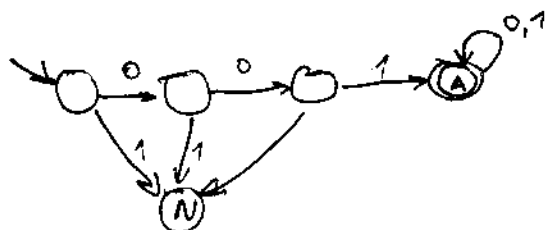
$$u_1 \leq u_2 \leq u_3 \leq \dots \leq u_n$$



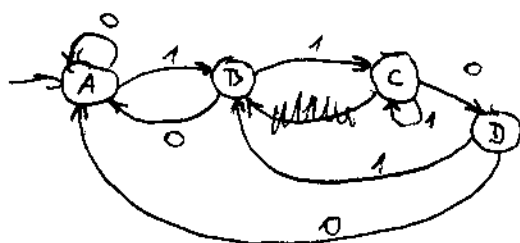
automaty

$A = \{0, 1\}$

slovo začínající 001



končící 110



~~stav~~ stav = A;

~~while~~

while (dalsipismeno) {

case stav {

A: if (pismeno == 1) stav = B;

B: ...

C: if (pismeno == 0) stav = D

D: ...

}

}

if stav == D print ('Konci 110')