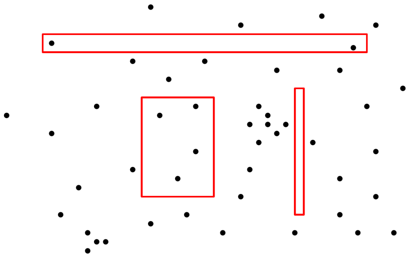
Geometria

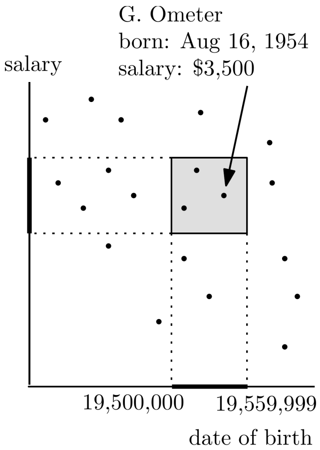
kuko

13.10.2020

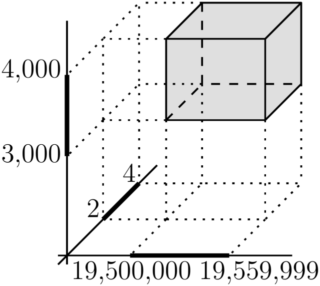
Vybrané partie z dátových štruktúr



A database query may ask for all employees with age between a_1 and a_2, and salary between s_1 and s_2



Example of a 3-dimensional (orthogonal) range query: children in $[2, 4]$, salary in $[3000, 4000]$, date of birth in $[19{,}500{,}000\ ,\ 19{,}559{,}999]$



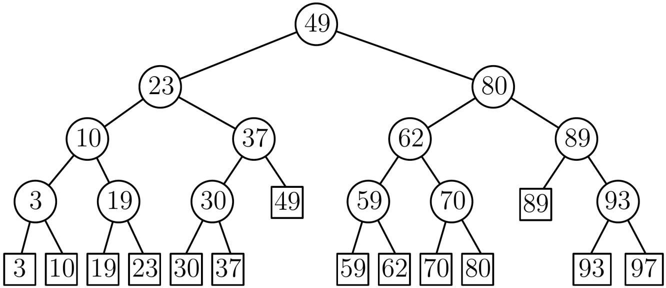
- dané: body $p_1, p_2, \ldots, p_n$
- môžeme si ich predspracovať
- chceme vedieť
  - spočítať body v danom obdĺžniku
  - vypísať všetky body v danom obdĺžniku
- dynamická úloha: body môžeme pridávať/mazať

- dané: body $p_1, p_2, \ldots, p_n$
- môžeme si ich predspracovať
- chceme vedieť
  - spočítať body v danom obdĺžniku
  - vypísať všetky body v danom obdĺžniku
- dynamická úloha: body môžeme pridávať/mazať

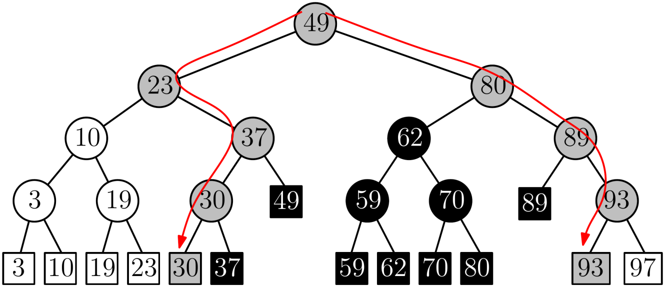
- dané: body $p_1, p_2, \ldots, p_n$
- môžeme si ich predspracovať
- chceme vedieť
  - spočítať body v danom obdĺžniku
  - vypísať všetky body v danom obdĺžniku
- dynamická úloha: body môžeme pridávať/mazať

1D ?

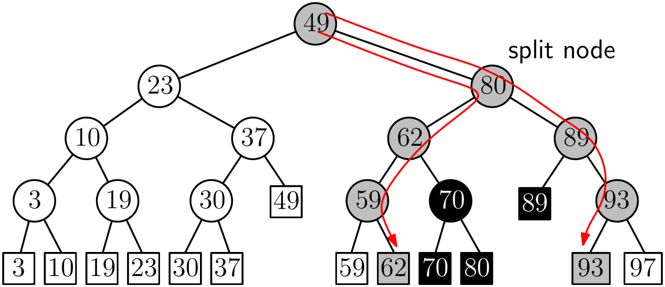
A balanced binary search tree with the points in the leaves



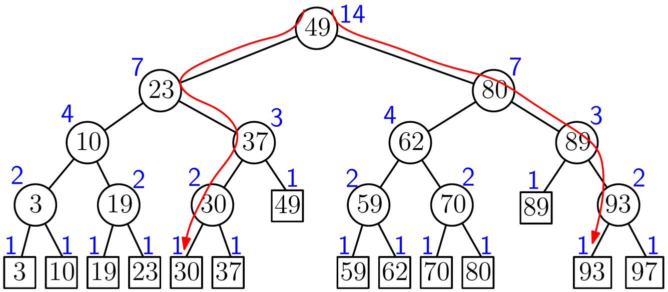
A 1-dimensional range query with $[25, 90]$



A 1-dimensional range query with $[61, 90]$



A 1-dimensional range counting query with $[25, 90]$



- počítanie v $O(\log n)$
- vypísanie k bodov v $O(\log n + k)$

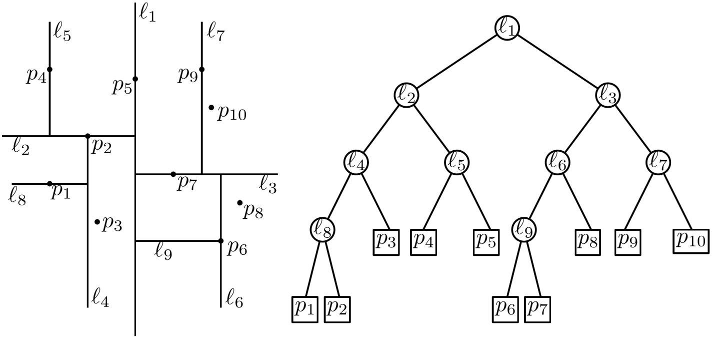
2D?

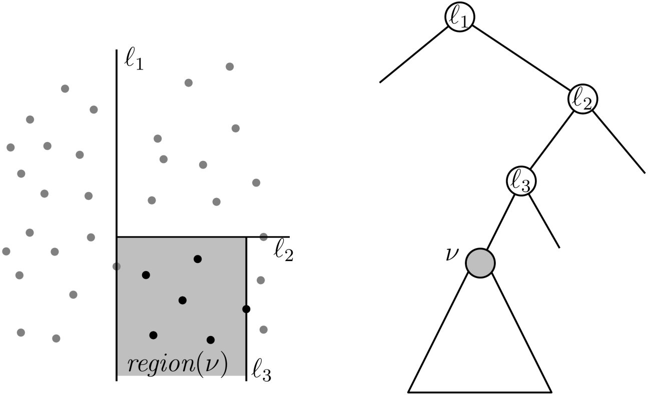
Kd-stromy: body budeme rozdeľovať striedavo podľa x-ových súradníc a y-ových súradníc

- podľa x: rozdelíme body podľa zvislej priamky tak, že polovica je vľavo a polovica vpravo
- podľa y: rozdelíme body podľa vodorovnej priamky tak, že polovica je dolu a polovica hore

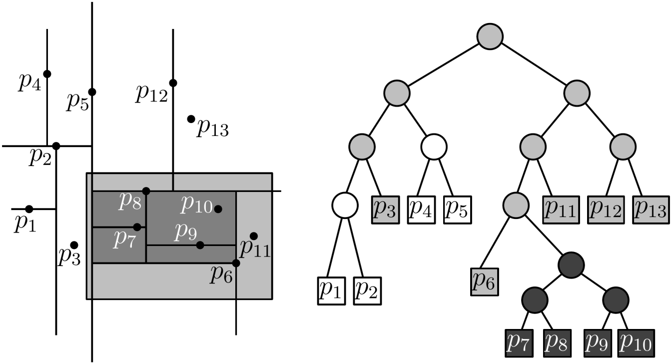
Kd-stromy: body budeme rozdeľovať striedavo podľa x-ových súradníc a y-ových súradníc

- podľa x: rozdelíme body podľa zvislej priamky tak, že polovica je vľavo a polovica vpravo
- podľa y: rozdelíme body podľa vodorovnej priamky tak, že polovica je dolu a polovica hore

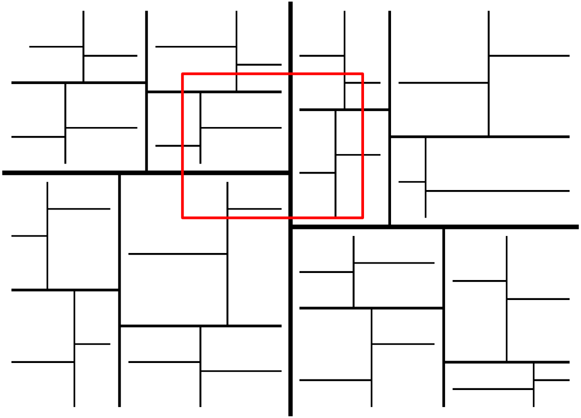

ℓ_1
ℓ_2
ℓ_3
ℓ_4
ℓ_5
ℓ_6
ℓ_7
ℓ_8
ℓ_9
p_1
p_2
p_3
p_4
p_5
p_6
p_7
p_8
p_9
p_{10}



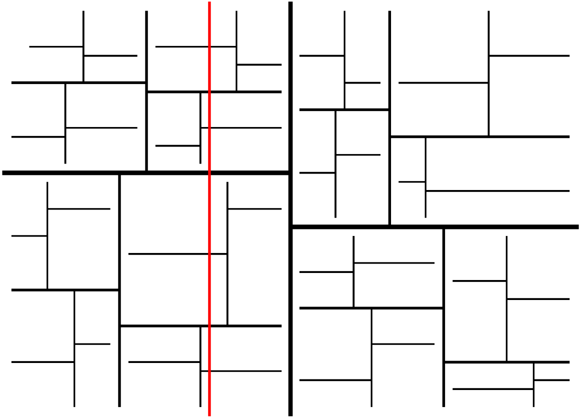
každý podstrom zodpovedá obdĺžnikovej oblasti



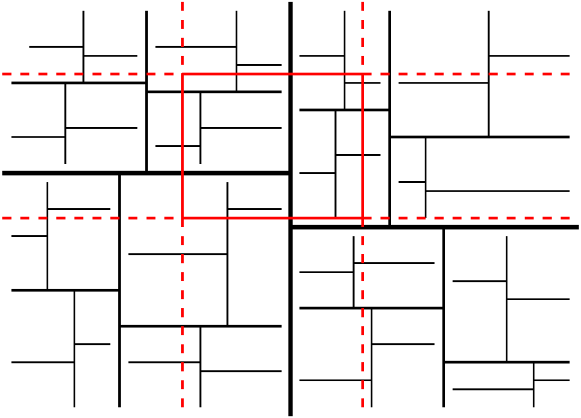
čierne vrcholy: celá oblasť je v obdĺžniku;
šedé: časť je v obdĺžniku



Question: How many grey and how many black *nodes*?



Question: How many grey and how many black *leaves*?



nech $G(n)$ je počet šedých vrcholov

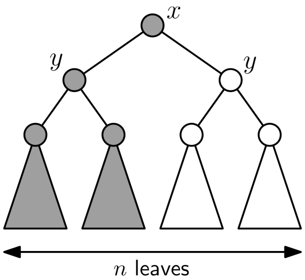
- $G(n) = G(n/2) + 1$ pre párne hĺbky
- $G(n) = 2 \cdot G(n/2) + 1$ pre nepárne hĺbky

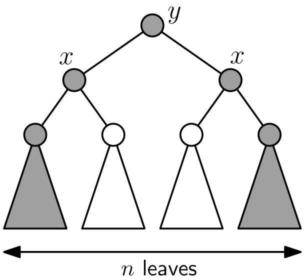
$$G(n) = 2 \cdot G(n/4) + O(1) \qquad \text{a} \qquad G(1) = 1$$

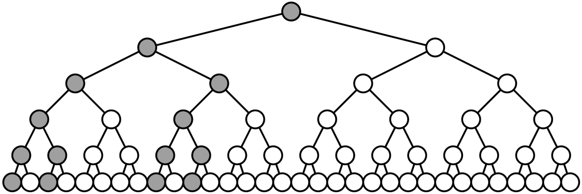
nech $G(n)$ je počet šedých vrcholov

- $G(n) = G(n/2) + 1$ pre párne hĺbky
- $G(n) = 2 \cdot G(n/2) + 1$ pre nepárne hĺbky

$$G(n) = 2 \cdot G(n/4) + O(1) \qquad \text{a} \qquad G(1) = 1$$


x
y
y
n leaves


y
x
x
n leaves



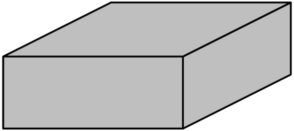
The grey subtree has unary and binary nodes

- počítanie v $O(\sqrt{n})$
- vypísanie k bodov v $O(\sqrt{n}+k)$

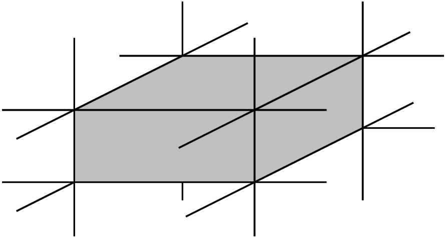
3D ?

A 3-dimensional kd-tree alternates splits on x-, y-, and z-coordinate

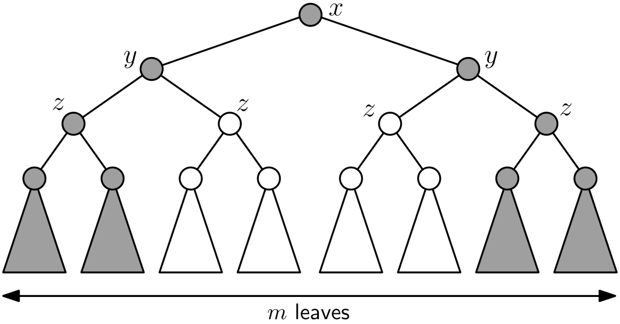
A 3D range query is performed with a box



How does the query time analysis change?



Intersection of B and $region(v)$ depends on intersection of facets of B $\Rightarrow$ analyze by axes-parallel planes (B has no more grey nodes than six planes)


x
y
y
z
z
z
z
m leaves

$$G(n) = 4 \cdot G(n/8) + O(1) \qquad \text{a} \qquad G(1) = 1$$

strom výšky $\frac{2}{3} \lg n$

$$G(n) = 4 \cdot G(n/8) + O(1) \qquad \text{a} \qquad G(1) = 1$$

strom výšky $\frac{2}{3} \lg n$

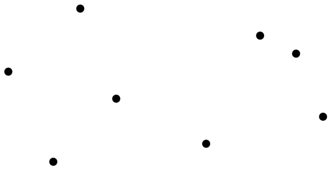
- počítanie v $O(n^{2/3})$
- v d rozmeroch $O(n^{1-1/d})$
- predpočítanie v $O(n \log n)$

- počítanie v $O(n^{2/3})$
- v d rozmeroch $O(n^{1-1/d})$
- predpočítanie v $O(n \log n)$

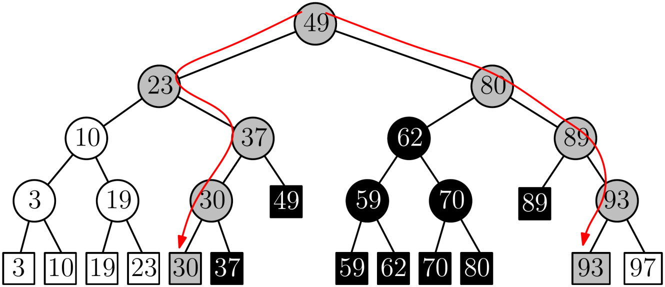
- počítanie v $O(n^{2/3})$
- v d rozmeroch $O(n^{1-1/d})$
- predpočítanie v $O(n \log n)$

Dá sa to lepšie?

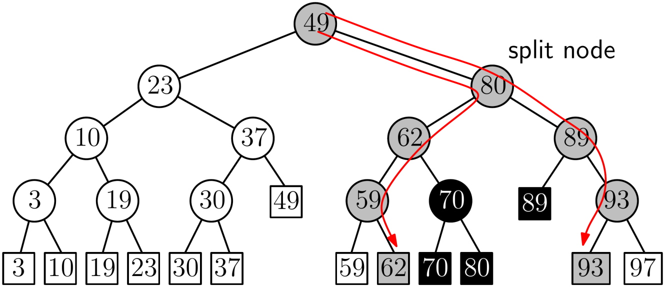
Can we achieve $O(\log n \, [+k])$ query time?



A 1-dimensional range query with $[25, 90]$

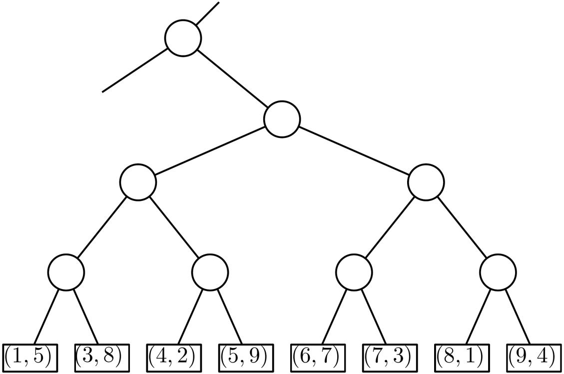


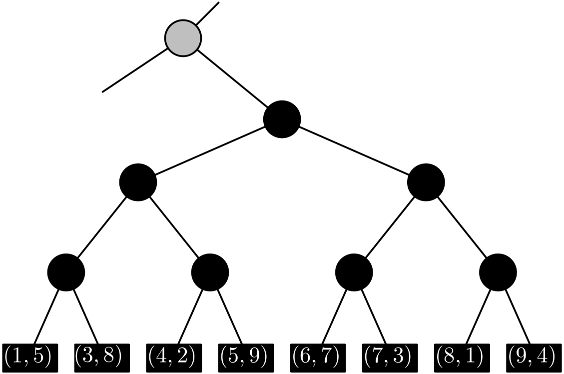
A 1-dimensional range query with $[61, 90]$

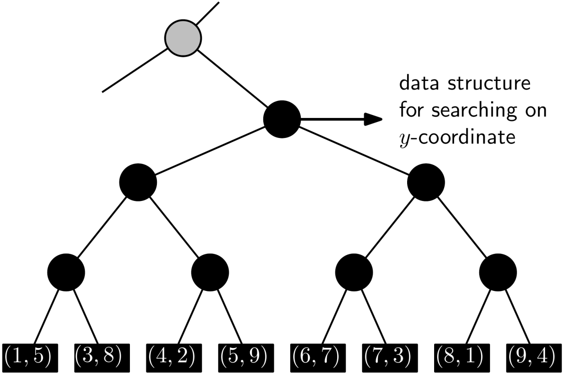


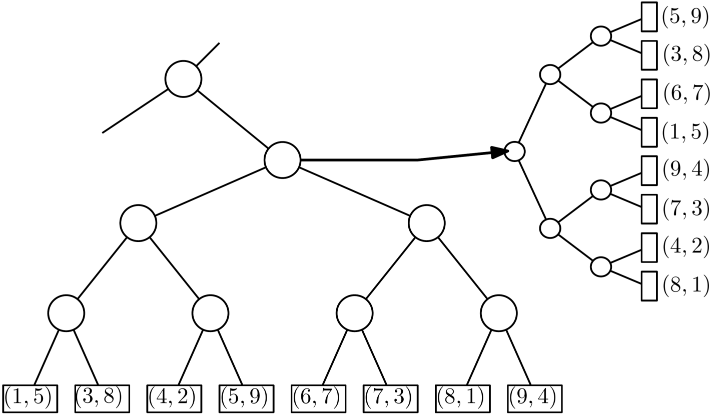
V 1D vieme identifikovať $O(\log n)$ podstromov, ktoré spolu reprezentujú celý rozsah.

V 2D vieme takto identifikovať $O(\log n)$ podstromov, ktoré obsahujú body *so správnou x-ovou súradnicou*.

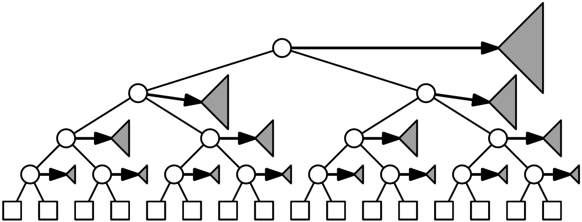

(1, 5)
(3, 8)
(4, 2)
(5, 9)
(6, 7)
(7, 3)
(8, 1)
(9, 4)


$(1,5)$
$(3,8)$
$(4,2)$
$(5,9)$
$(6,7)$
$(7,3)$
$(8,1)$
$(9,4)$


data structure
for searching on
y-coordinate
$(1,5)$
$(3,8)$
$(4,2)$
$(5,9)$
$(6,7)$
$(7,3)$
$(8,1)$
$(9,4)$


(1, 5)
(3, 8)
(4, 2)
(5, 9)
(6, 7)
(7, 3)
(8, 1)
(9, 4)
(5, 9)
(3, 8)
(6, 7)
(1, 5)
(9, 4)
(7, 3)
(4, 2)
(8, 1)

Every internal node stores a whole tree in an *associated structure*, on y-coordinate

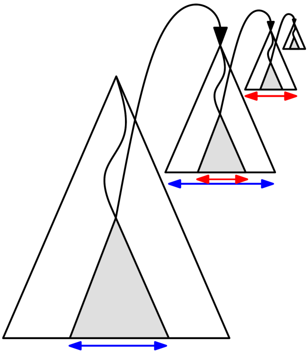


Question: How much storage does this take?

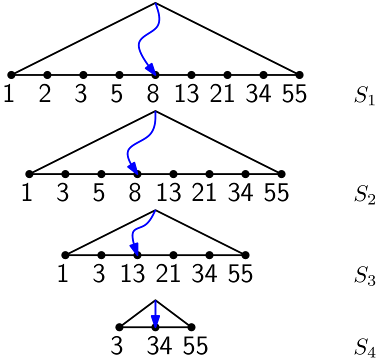
- pamäť $O(n \log n)$
- predspracovanie v $O(n \log n)$
- query: $O(\log^2 n)$

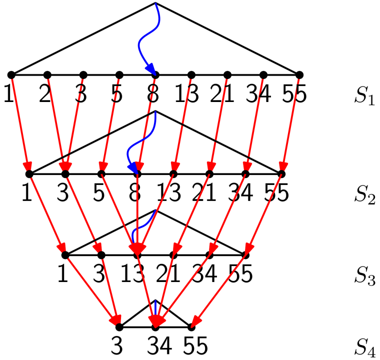
n	$\log n$	$\log^2 n$	$\sqrt{n}$
16	4	16	4
64	6	36	8
256	8	64	16
1024	10	100	32
4096	12	144	64
16384	14	196	128
65536	16	256	256
1M	20	400	1K
16M	24	576	4K

A d-dimensional range tree has a main tree which is a one-dimensional balanced binary search tree on the first coordinate, where every node has a pointer to an associated structure that is a $(d-1)$-dimensional range tree on the other coordinates

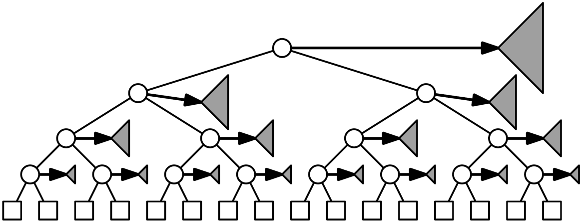


Dá sa to lepšie?

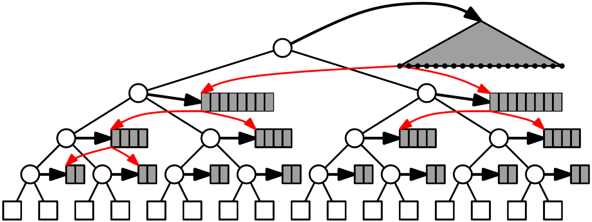

1 2 3 5 8 13 21 34 55
S_1
1 3 5 8 13 21 34 55
S_2
1 3 13 21 34 55
S_3
3 34 55
S_4


1 2 3 5 8 13 21 34 55
1 3 5 8 13 21 34 55
1 3 13 21 34 55
3 34 55
S_1
S_2
S_3
S_4

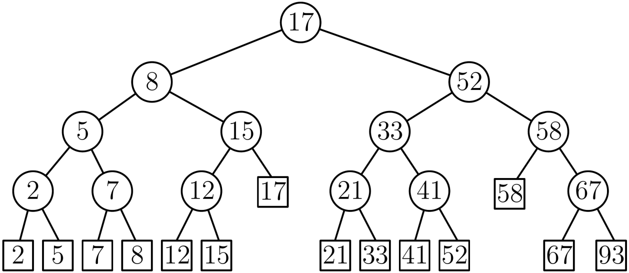
Every internal node stores a whole tree in an *associated structure*, on y-coordinate



Question: How much storage does this take?

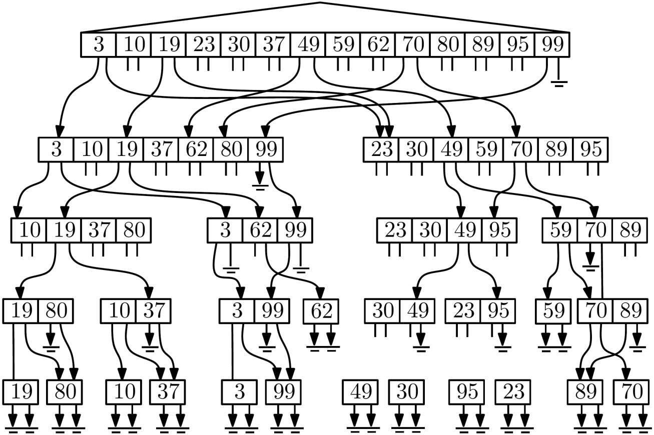


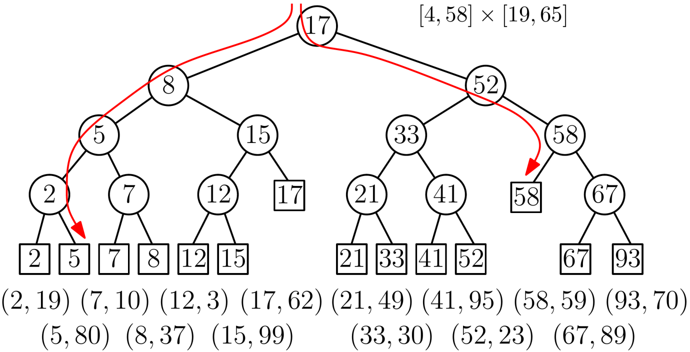
Napríklad vstup:
$(2,19),(5,80),(7,10),(8,37),(12,3),(15,99),(17,62)$, $(21,49)$, $(33,30)$, $(41,95),(52,23),(58,59),(67,89),(93,70)$

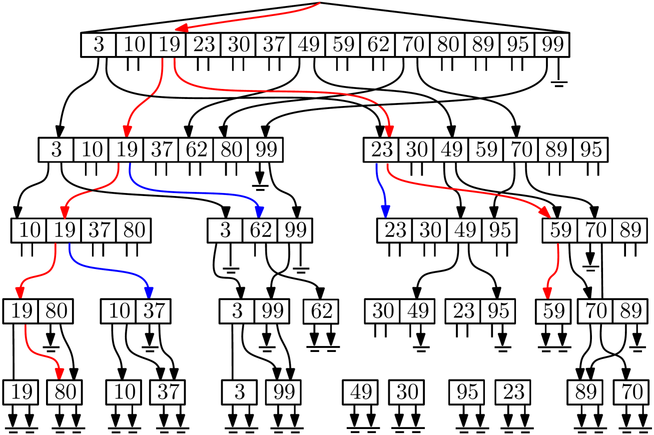


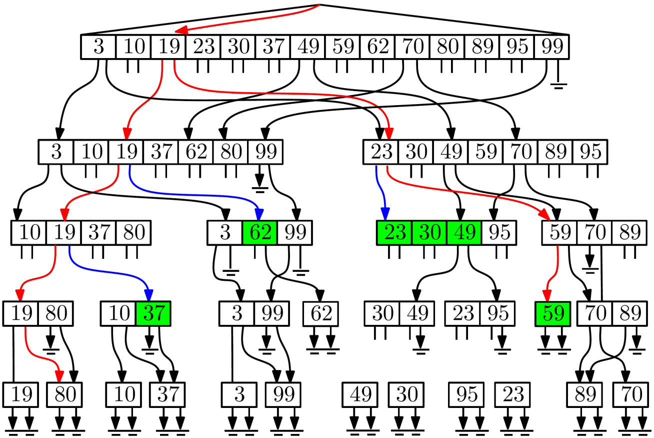
$(2, 19)$ $(7, 10)$ $(12, 3)$ $(17, 62)$ $(21, 49)$ $(41, 95)$ $(58, 59)$ $(93, 70)$

$(5, 80)$ $(8, 37)$ $(15, 99)$ $(33, 30)$ $(52, 23)$ $(67, 89)$


3 10 19 23 30 37 49 59 62 70 80 89 95 99
3 10 19 37 62 80 99
23 30 49 59 70 89 95
10 19 37 80
3 62 99
23 30 49 95
59 70 89
19 80
10 37
3 99
62
30 49
23 95
59
70 89
19
80
10
37
3
99
49
30
95
23
89
70


[4, 58] × [19, 65]
17
8
52
5
15
33
58
2
7
12
17
21
41
58
67
2
5
7
8
12
15
21
33
41
52
67
93
(2, 19) (7, 10) (12, 3) (17, 62) (21, 49) (41, 95) (58, 59) (93, 70)
(5, 80) (8, 37) (15, 99) (33, 30) (52, 23) (67, 89)


3 10 19 23 30 37 49 59 62 70 80 89 95 99
3 10 19 37 62 80 99
23 30 49 59 70 89 95
10 19 37 80
3 62 99
23 30 49 95
59 70 89
19 80
10 37
3 99
62
30 49
23 95
59
70 89
19
80
10
37
3
99
49
30
95
23
89
70


3 10 19 23 30 37 49 59 62 70 80 89 95 99
3 10 19 37 62 80 99
23 30 49 59 70 89 95
10 19 37 80
3 62 99
23 30 49 95
59 70 89
19 80
10 37
3 99
62
30 49
23 95
59
70 89
19
80
10
37
3
99
49
30
95
23
89
70

2D:

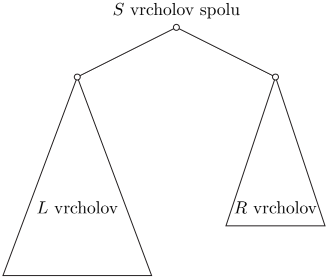
- pamäť $O(n \log n)$
- predspracovanie v $O(n \log n)$
- počítanie v $O(\log n)$

d rozmerov:

- pamäť $O(n \log^{d-1} n)$
- predspracovanie v $O(n \log^{d-1} n)$
- počítanie v $O(\log^{d-1} n)$

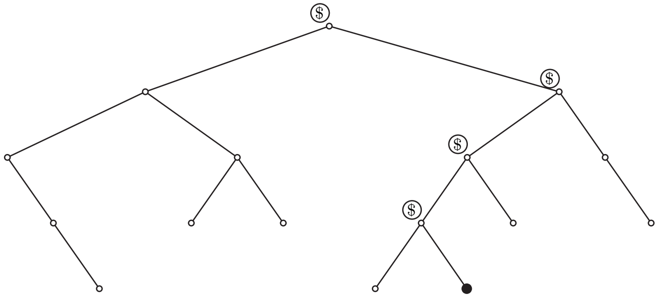
Dynamicky?

Pamätáte si scapegoat stromy?



$$\Delta = |L - R|$$

INVARIANT: Každý vrchol bude mať našetrené aspoň $\Delta - 1$\$.



Na každý `insert` stačí $2 \log_{3/2} N\$$:

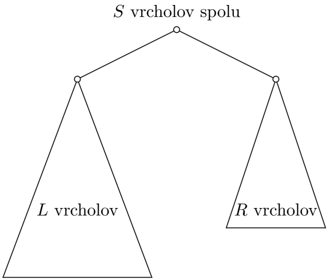
- $\log_{3/2} N\$$ zaplatí pridanie
- $\log_{3/2} N\$$ si ušetríme na neskôr

…tým pádom nevyvážený podstrom s S vrcholmi mal $\Theta(S)$\$ a vedel zaplatiť prebudovanie v $O(S)$ **Problém:** tentokrát trvá

prebudovanie $O(S \log S)$ (v 2D)

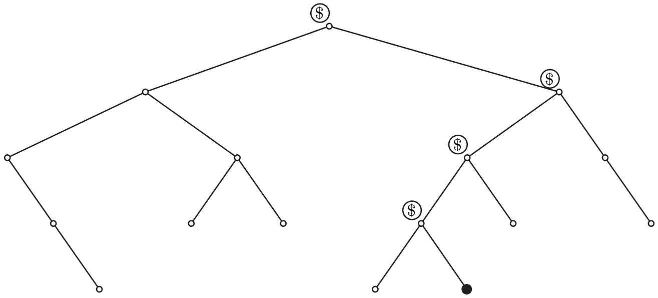
...tým pádom nevyvážený podstrom s S vrcholmi mal $\Theta(S)$\$ a vedel zaplatiť prebudovanie v $O(S)$ **Problém:** tentokrát trvá

prebudovanie $O(S \log S)$ (v 2D)



$$\Delta = |L - R|$$

INVARIANT: Každý vrchol bude mať našetrené aspoň $(\Delta - 1)\log S$\$.



Na každý `insert` stačí $O(\log^2 N)\$$:

- $\log N\$$ zaplatí pridanie
- každý vrchol si ušetrí $\approx \log N\$$ na neskôr

2D:

- pamäť $O(n \log n)$
- predspracovanie v $O(n \log n)$
- počítanie v $O(\log n)$
- insert/delete v $O(\log^2 n)$ amortizovane