

Ännu mera träd:

2-3-träd, B-träd,
rödsvarta träd, 2-3-4-träd

Weíss, avsnitt 4.7, 11.5, 12.2, etc.

Peter Ljunglöf

DAT036, Datastrukturer

30 nov 2012

2-3-träd

[inte i kursboken]

Ett 2-3-träd har två sorters noder: 2-noder och 3-noder

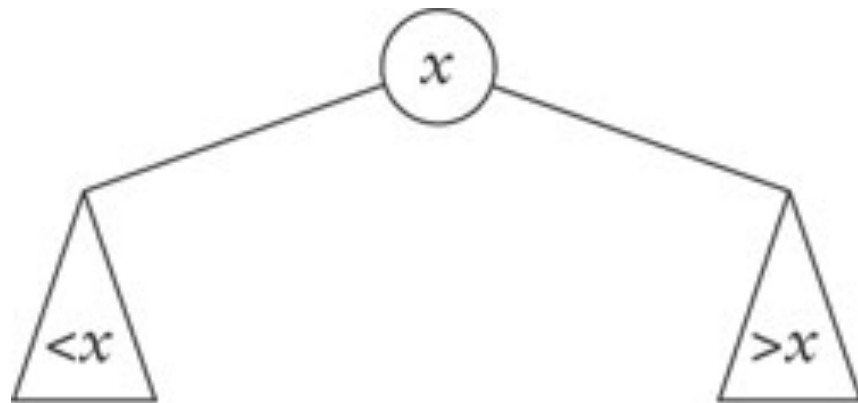
En 2-nod är en vanlig binär sökträdsnod:

- den innehåller ett datafält (d_1) och två barn (b_1, b_2)
- det första barnets (b_1) data är mindre än nodens (d_1)
- det andra barnets (b_2) data är större än nodens (d_1)

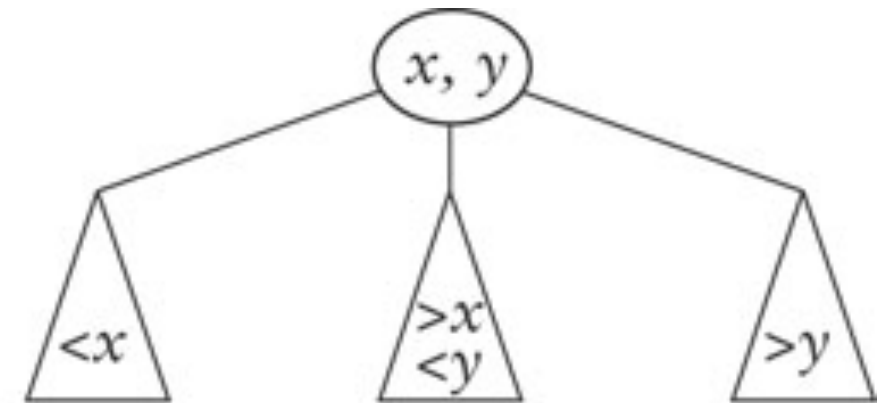
En 3-nod har två datafält (d_1, d_2) och tre barn (b_1, b_2, b_3):

- det första datafältet (d_1) är mindre än det andra (d_2)
- det första barnets (b_1) data är mindre än d_1
- det andra barnets (b_2) data ligger mellan d_1 och d_2
- det tredje barnets (b_3) data är större än d_2

2-3-träd

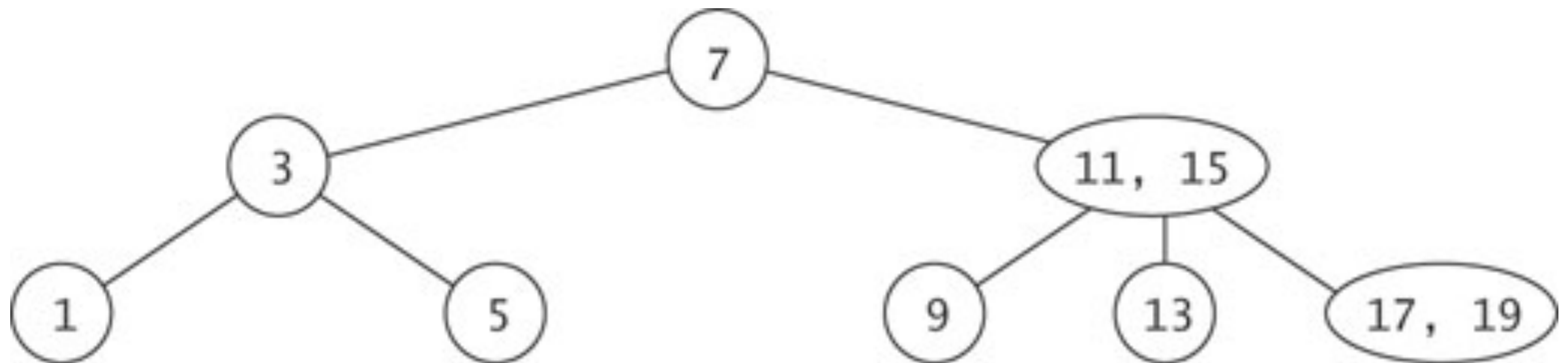


2-node



3-node

Ett exempel på ett 2-3-träd:



Ett 2-3-exempel

Nu ska vi bygga ett 2-3-träd för orden i

"The quick brown fox jumps over the lazy dog"

The...

The

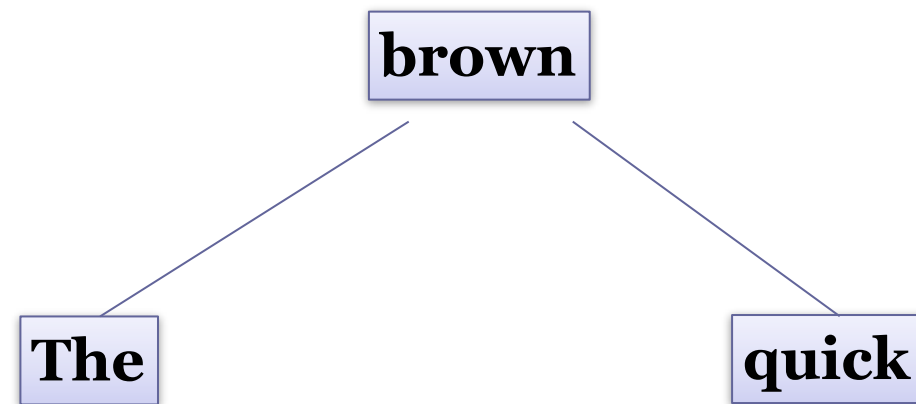
The quick...

The, quick

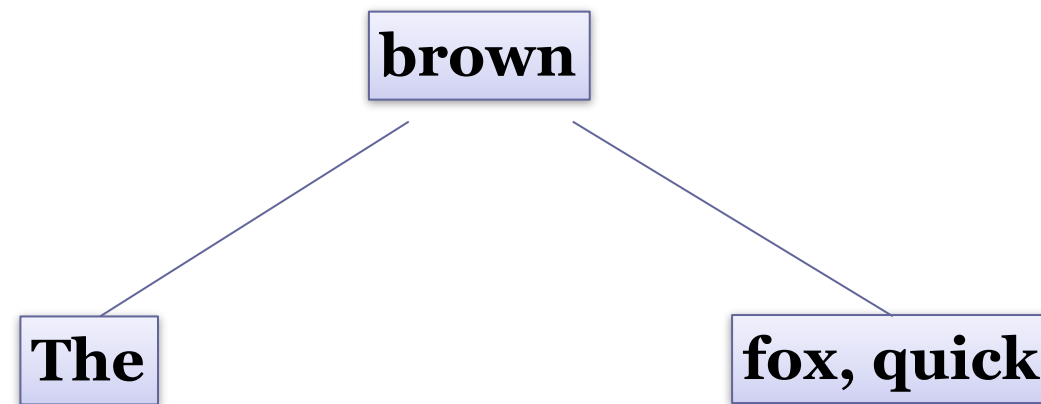
The quick brown...

The, brown, quick

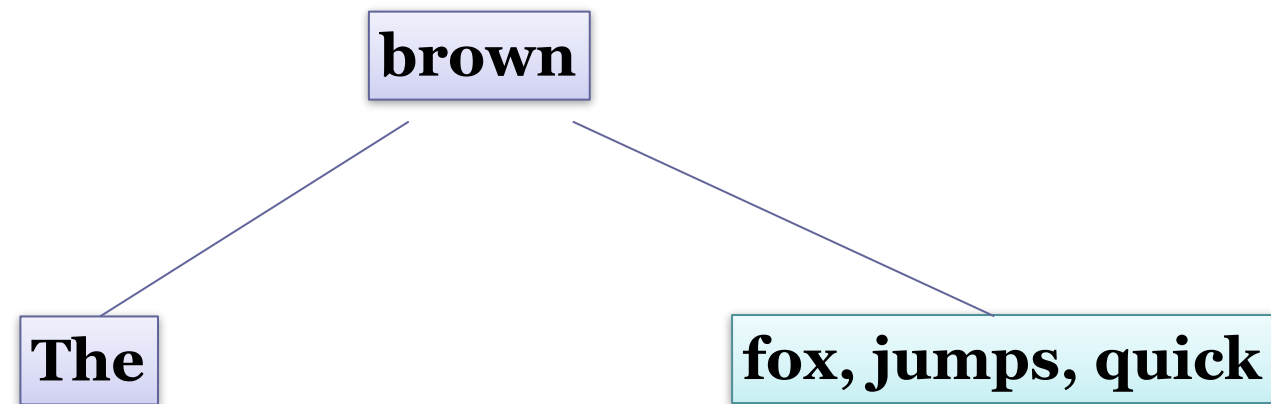
The quick brown...



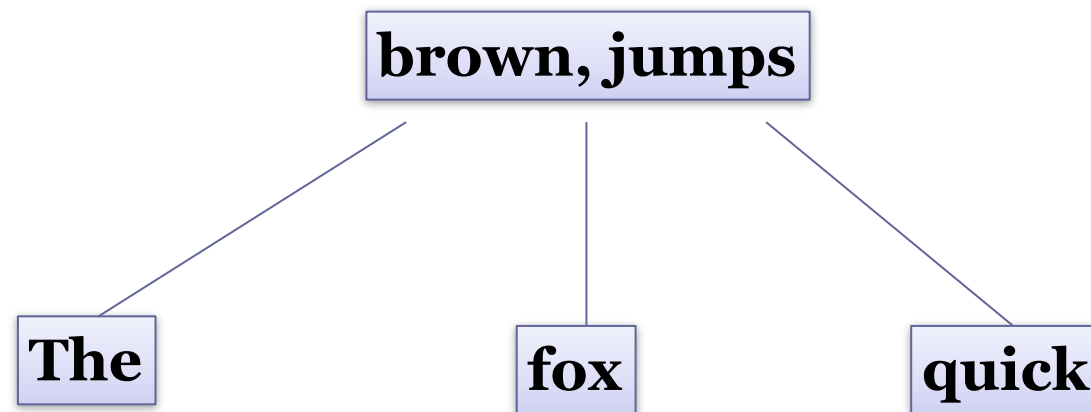
The quick brown fox...



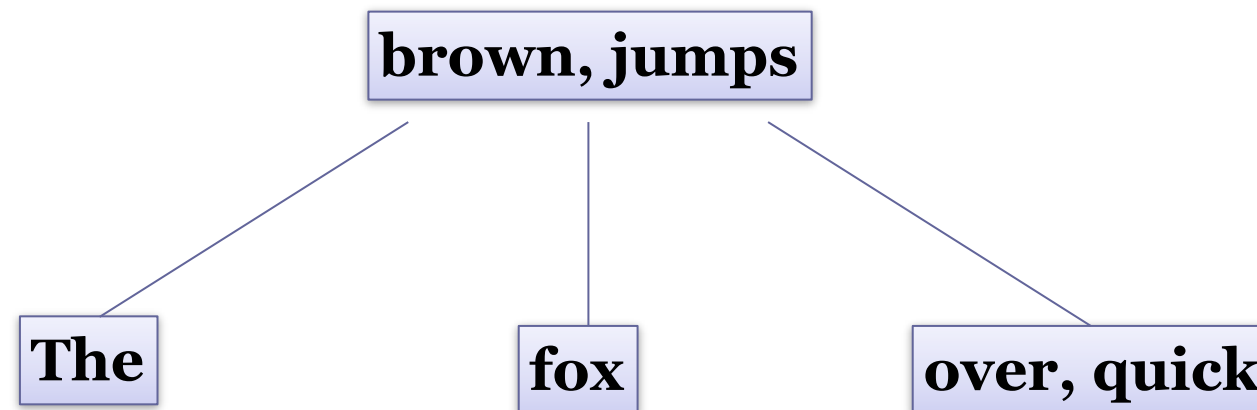
The quick brown fox jumps...



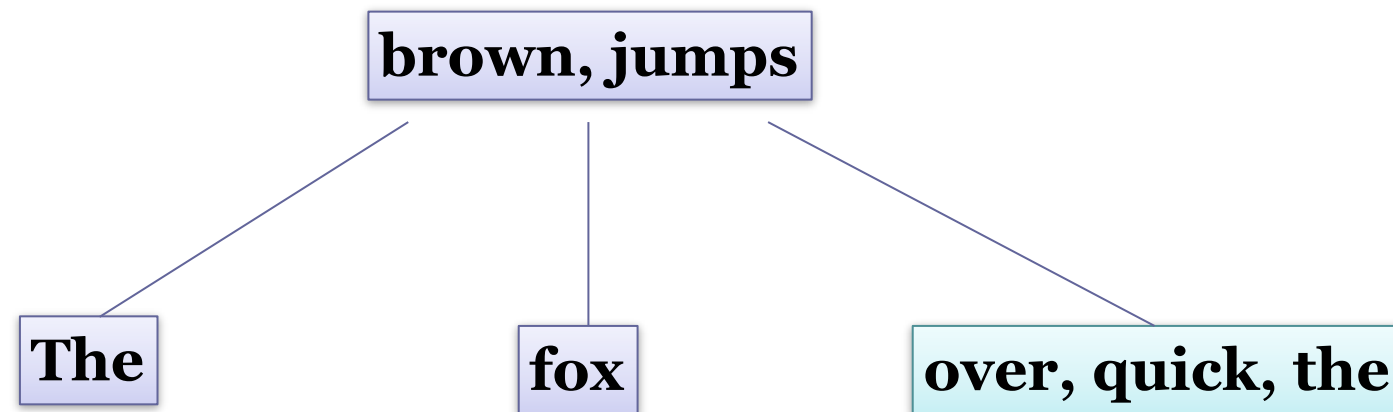
The quick brown fox jumps...



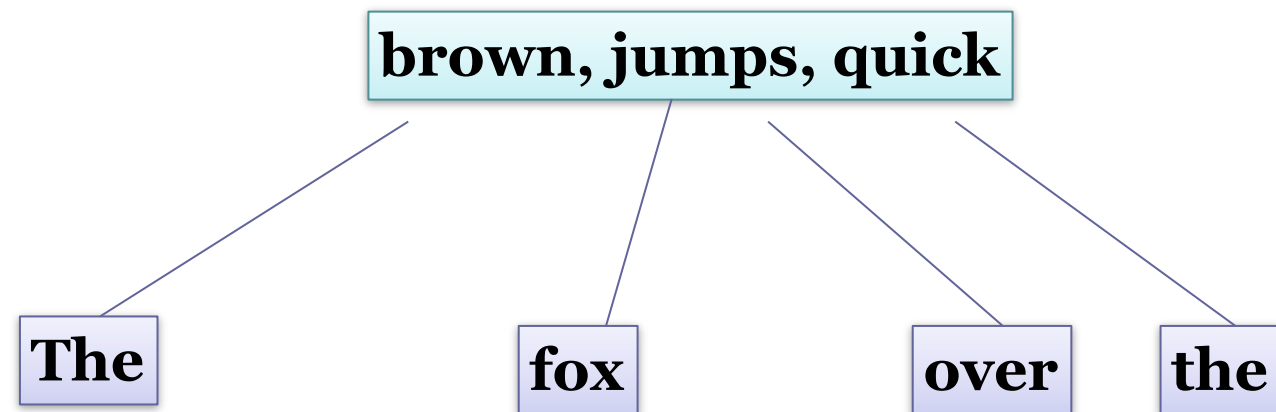
The quick brown fox jumps
over...



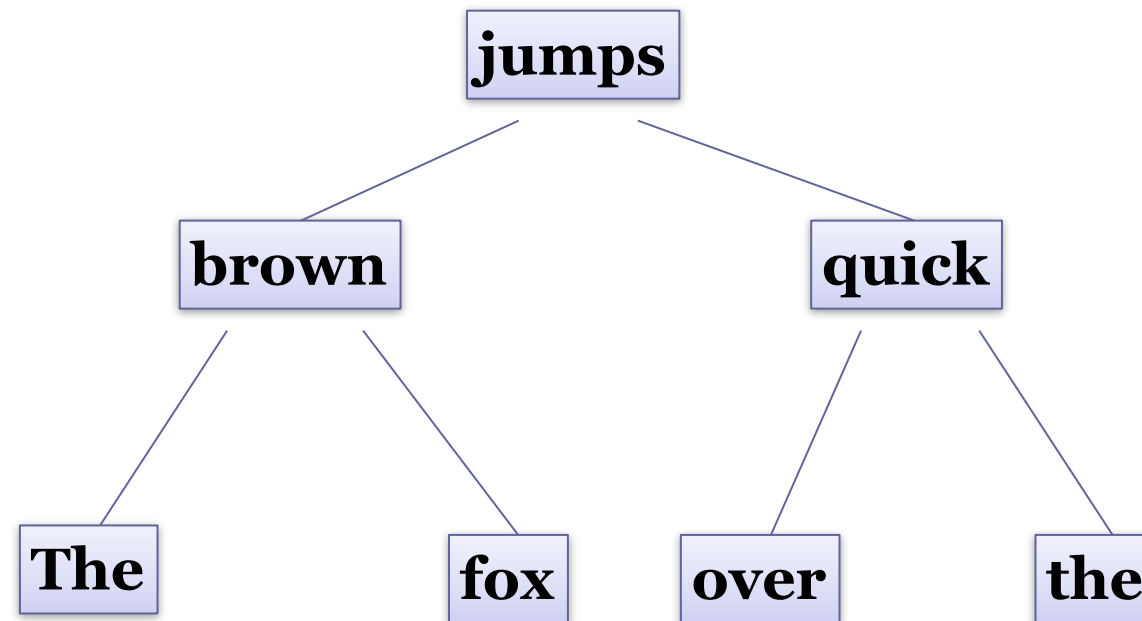
The quick brown fox jumps over the...



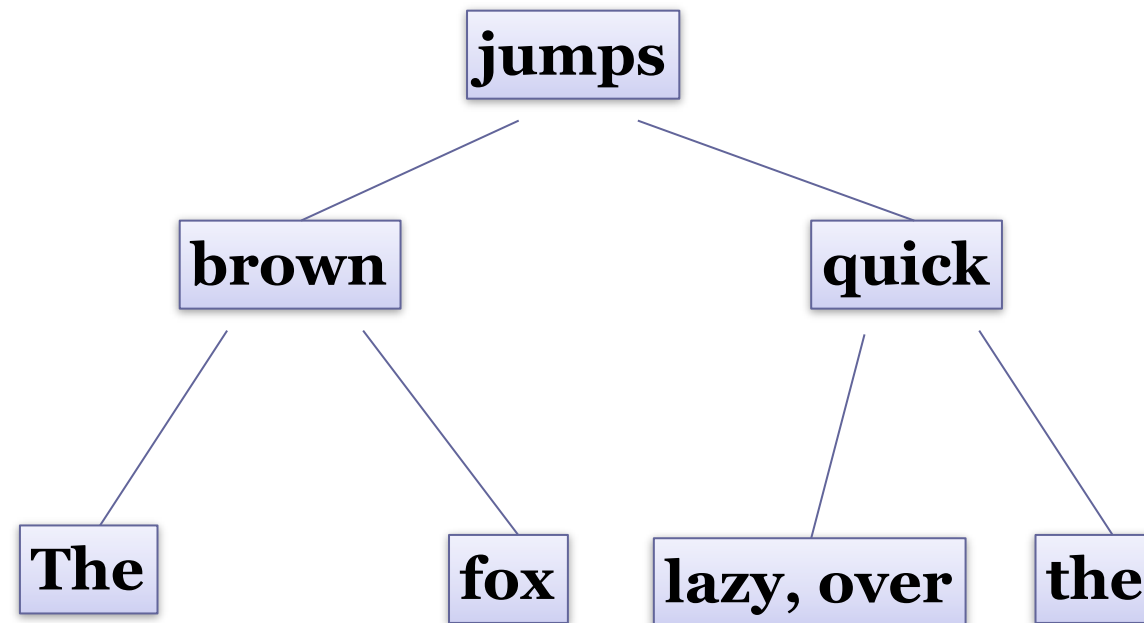
The quick brown fox jumps over the...



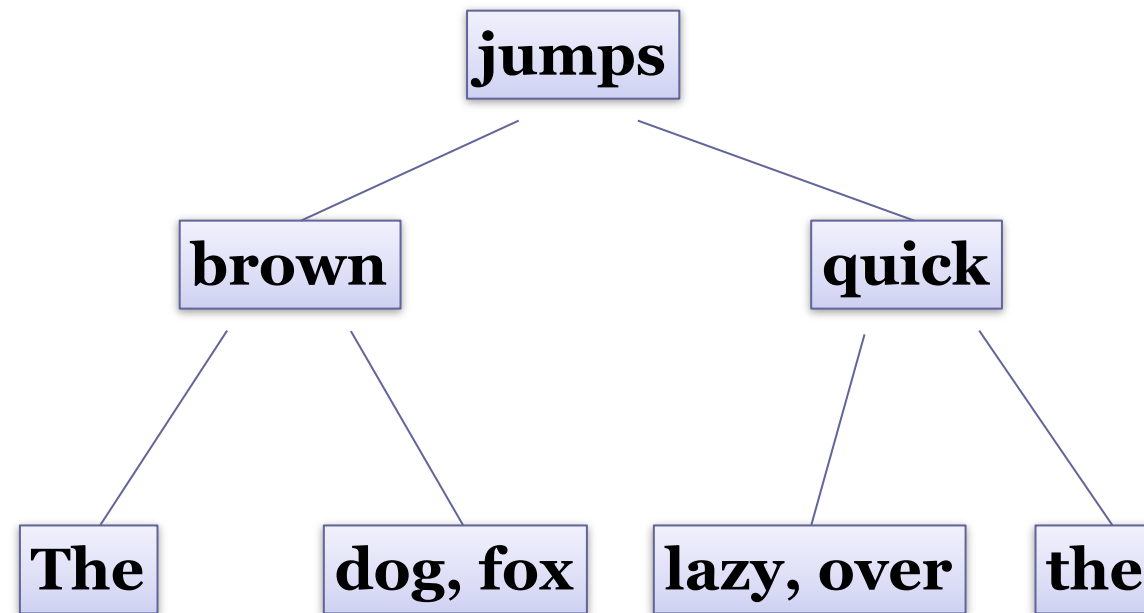
The quick brown fox jumps over the...



quick brown fox jumps over the lazy...



ck brown fox jumps over the lazy dog!



Analys av 2-3-träd

2-3-träd använder inte rotationer

- i motsats till AVL-träd
- istället ”bubblar” noderna uppåt genom trädet

Komplexiteten:

- antal noder n som får plats i ett 2-3-träd med höjd h är mellan 2^h (om alla noder är 2-noder) och 3^h (om alla noder är 3-noder)
- alltså, höjden h ligger mellan $\log_3 n$ och $\log_2 n$
- söktiden är $O(\log n)$

B-träd

[avsnitt 4.7]

B-träd är en generalisering av 2-3-träd:

- i ett B-träd av ordning k , kan varje nod ha upp till k barn

B-träd är designade för att lagra index till stora databaser:

- hårddiskars diskutrymme är uppdelade i block
- B-trädets ordning är precis sådan att en nod får plats i ett block
- (detta för att minimera antalet läsningar från hårddisken)

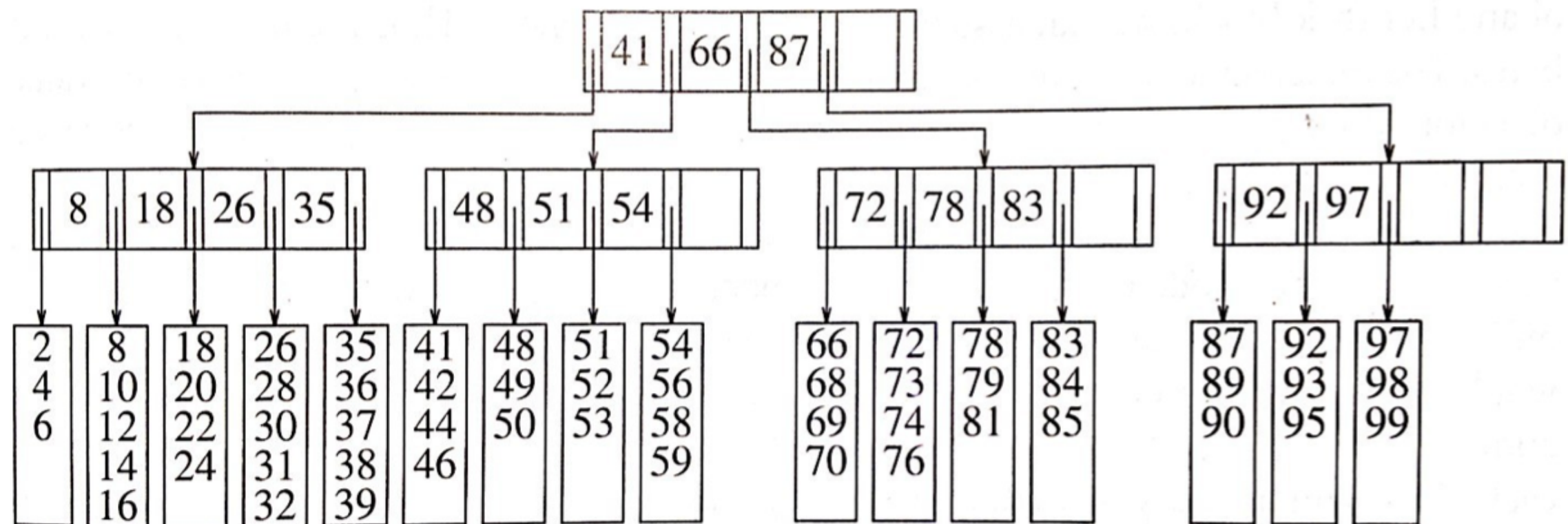


Figure 4.59 B-tree of order 5

Insättning i B-träd

Nya element stoppas in i ett passande löv.

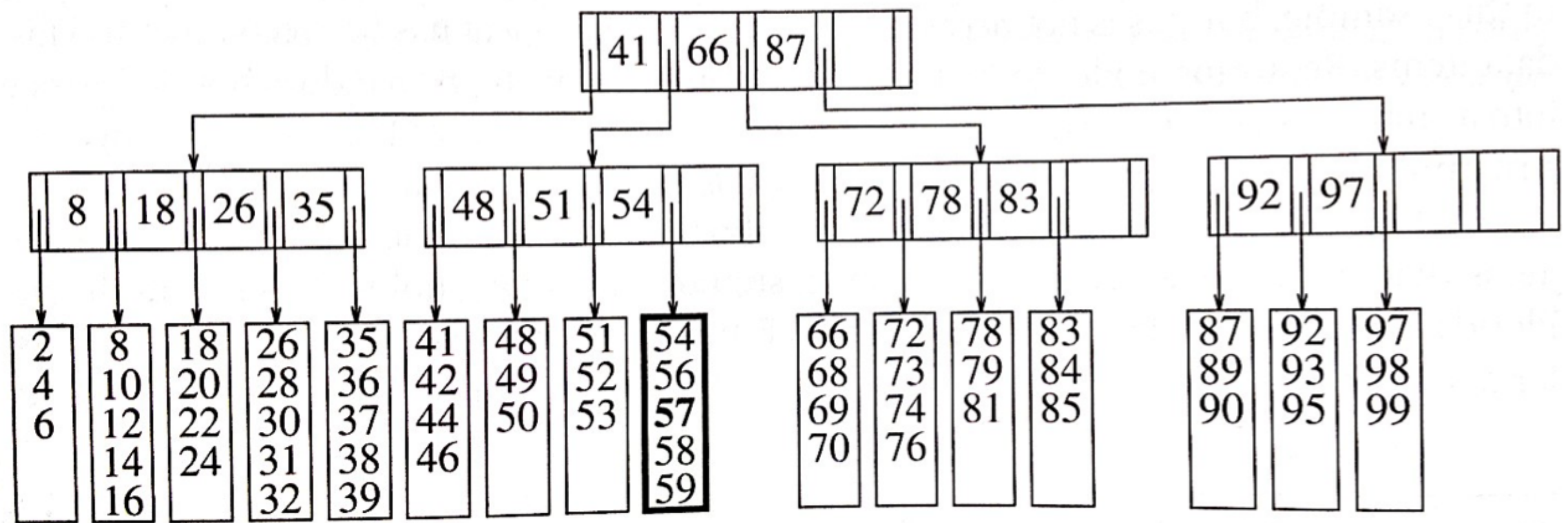


Figure 4.60 B-tree after insertion of 57 into the tree in Figure 4.59

Insättning i B-träd

Blir det för många element i ett löv så delar vi upp det i två.

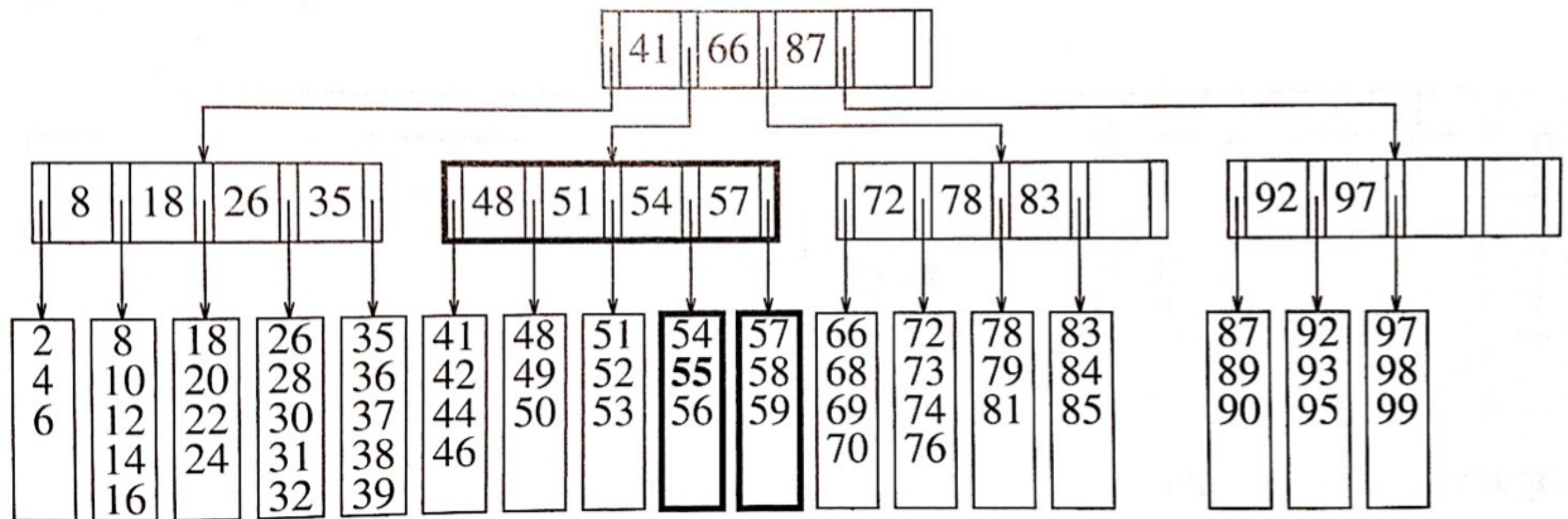


Figure 4.61 Insertion of 55 into the B-tree in Figure 4.60 causes a split into two leaves

Insättning i B-träd

Blir föräldranoden full så delar vi den också.

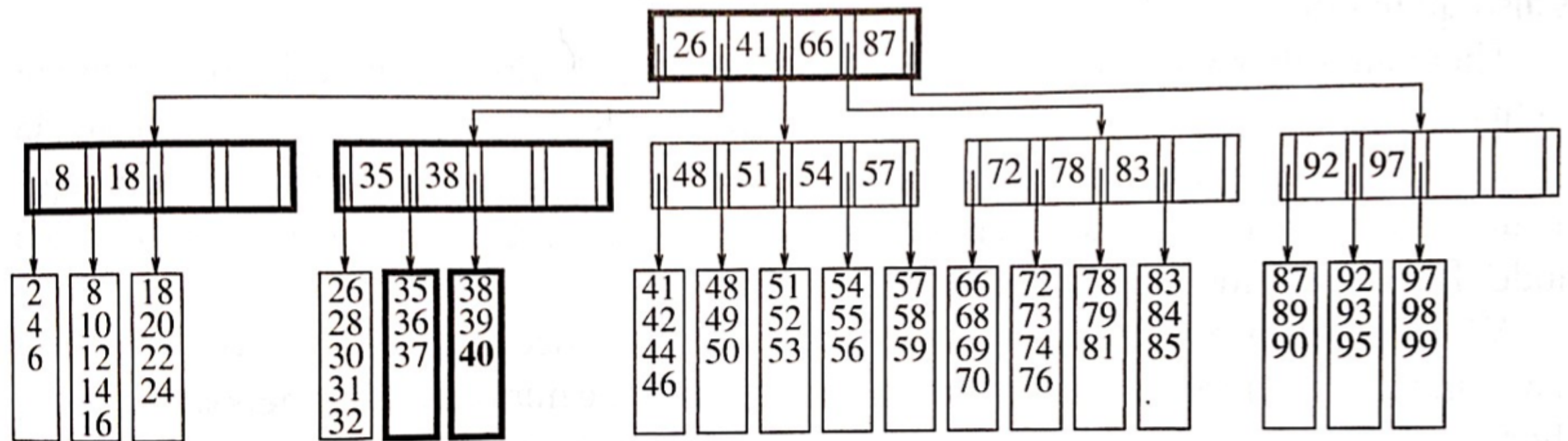


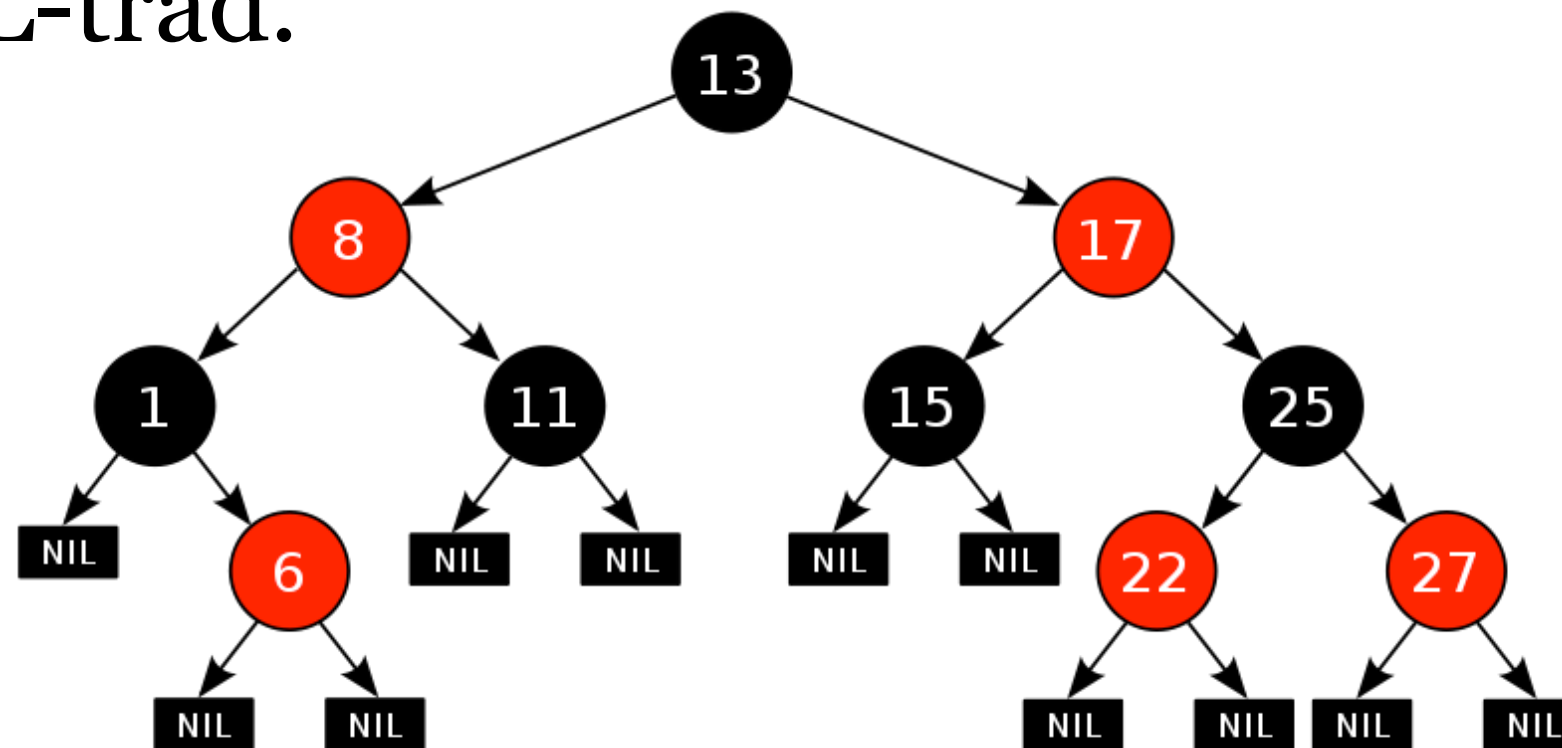
Figure 4.62 Insertion of 40 into the B-tree in Figure 4.61 causes a split into two leaves and then a split of the parent node

Rödsvarta träd

[avsnitt 12.2]

I ett rödsvart träd har alla noder en ”färg”:

- Antingen röd eller svart.
- Färgerna används för att kunna hålla trädet balanserat, precis som nodbalansen i ett AVL-träd.



Rödsvarta träd, invarianter

Ett rödsvart träd har följande invarianter:

- (1) en nod är antingen röd eller svart
- (2) roten är alltid svart
- (3) en röd nod har alltid svarta barn
- (4) tomma barn kallas "löv" och är svarta
- (5) antalet svarta noder är alltid samma,
i varje stig från roten till ett löv

(5) + (3) medför att antalet röda noder i en stig aldrig är fler än antalet svarta

☉ därför är trädet balanserat

Insättning i ett rödsvart träd

Wikipedias artikel om rödsvarta träd är bra!

http://en.wikipedia.org/wiki/Red-black_tree

Först söker vi efter insättningspunkten precis som för alla binära sökträd.

Det nya elementet ersätter ett löv och ges färgen röd:

- den nya noden får två svarta löv, så antalet svarta noder i lövstigarna är oförändrat
- om föräldern är svart, så är vi klara (Wikipedias fall 2)
- annars behöver vi arrangera om trädet
- det finns tre möjliga fall som vi måste hantera (Wikipedias fall 3, 4, 5)

Insättning, de olika fallen

Fall 2 i Wikipedia:

- föräldern är svart

Fall 3 i Wikipedia:

- både föräldern och dess syskon är röda

Fall 4 i Wikipedia:

- föräldern är röd och dess syskon svart
- dessutom är noden ett höger-vänsterbarn (eller ett vänster-högerbarn)

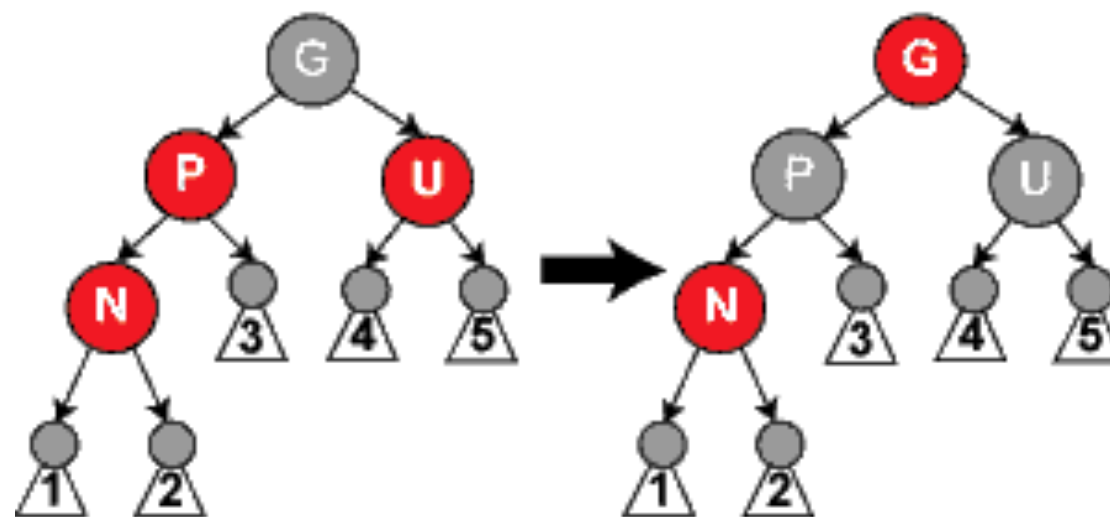
Fall 5 i Wikipedia:

- föräldern är röd och dess syskon svart
- dessutom är noden ett vänster-vänsterbarn (eller ett höger-högerbarn)

Insättning, fall 3

Både föräldern (P) och dess syskon (U) är röda:

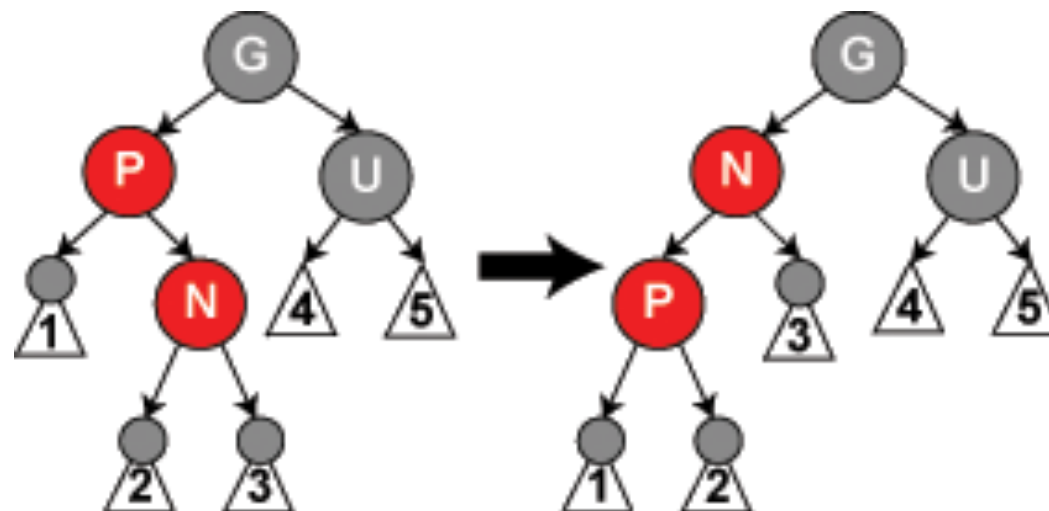
- deras förälder (G) måste då vara svart
- byt färg på dem (P, U) till svart
- byt färg på deras förälder (G) till röd
- fortsätt rekursivt uppåt i trädet med G som ny nod



Insättning, fall 4

Föräldern (P) är röd och dess syskon (U) är svart:

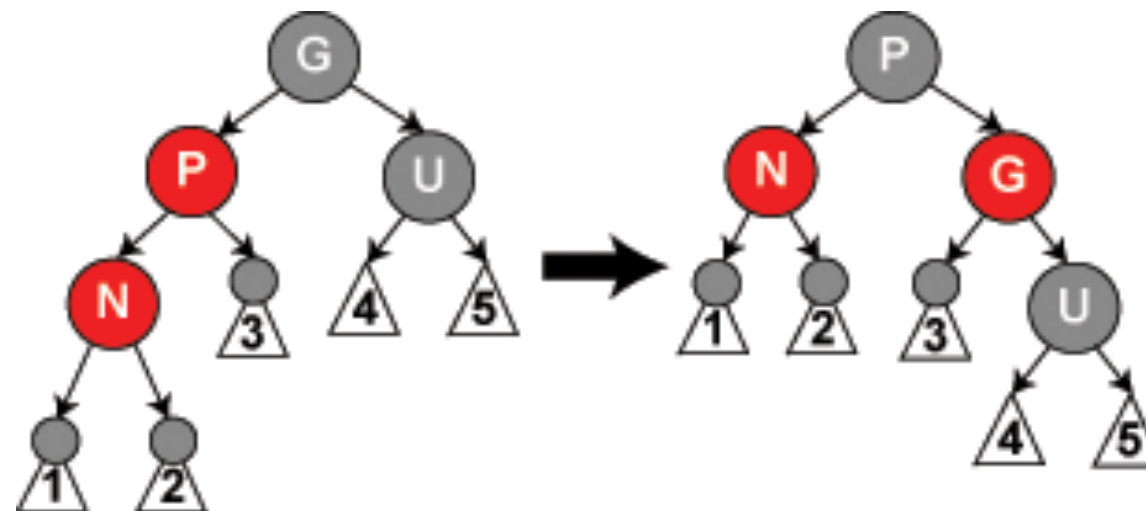
- dessutom är noden (N) ett vänster-högerbarn (eller ett höger-vänsterbarn)
- deras förälder (G) måste vara svart
- rotera runt P
- nu är P ett vänster-vänsterbarn (eller höger-högerbarn) och vi kan fortsätta med fall 5



Insättning, fall 5

Föräldern (P) är röd och dess syskon (U) är svart:

- dessutom är noden (N) ett vänster-vänsterbarn (eller ett höger-högerbarn)
- deras förälder (G) måste vara svart
- rotera runt G
- byt färg på P till svart och på G till röd
- nu är P ny svart lokal rot, och vi är klara!



Effektivitet hos rödsvarta träd

Höjden på ett rödsvart träd har en övre gräns:

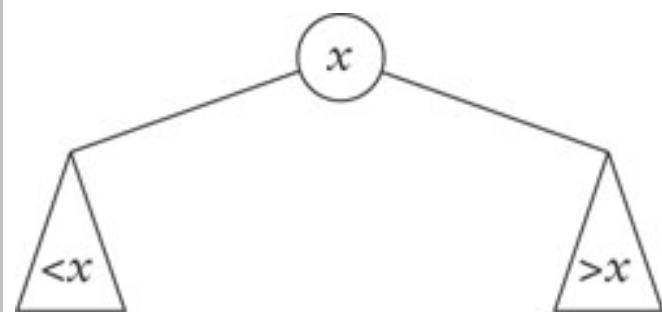
- maximalt är höjden $2 \cdot \log_2 n + 2$, vilket är logaritmiskt, $O(\log n)$
- precis som med AVL-träd så är det i medeltal mycket bättre än så

Javas API har klasserna TreeMap och TreeSet som är implementerade med rödsvarta träd

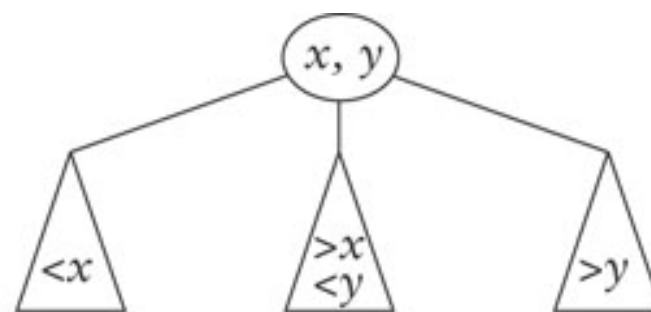
2-3-4-träd

[ej i kursboken]

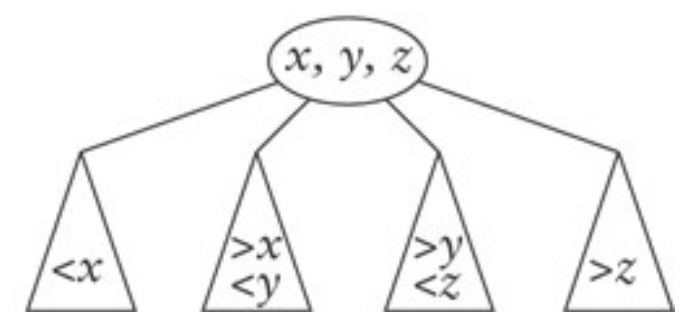
2-3-4-träd är B-träd av ordning 4



2-node

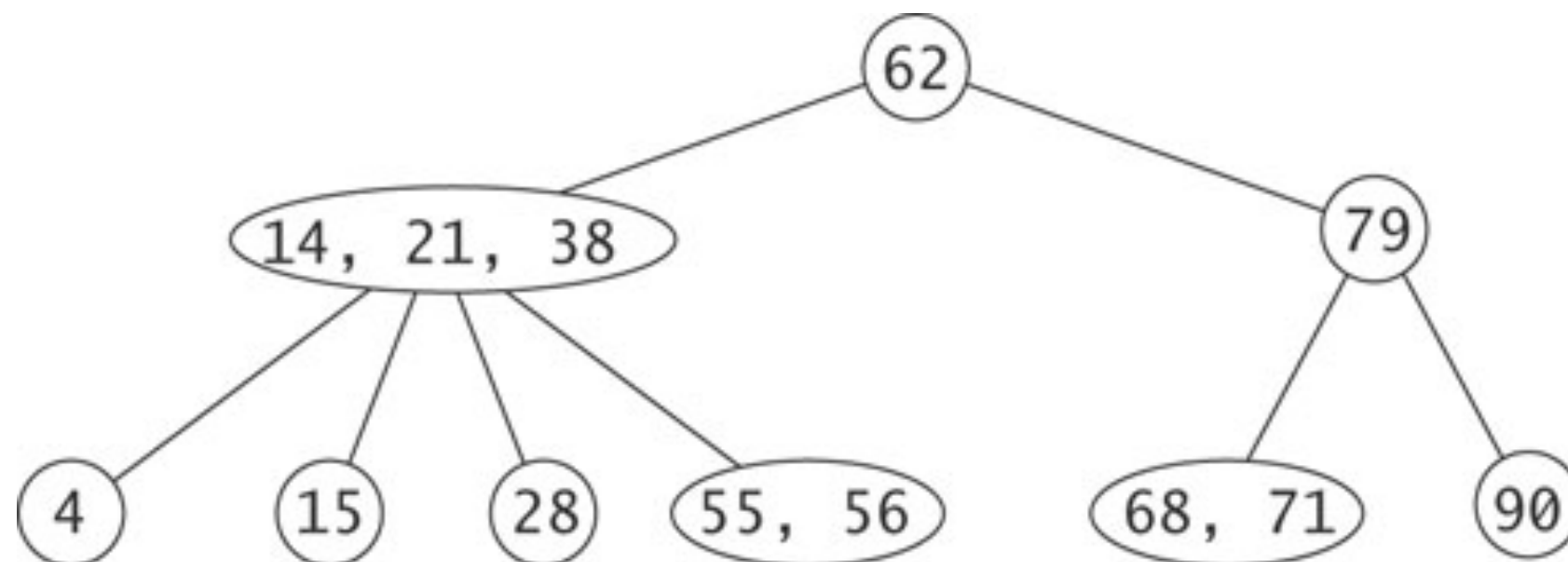


3-node



4-node

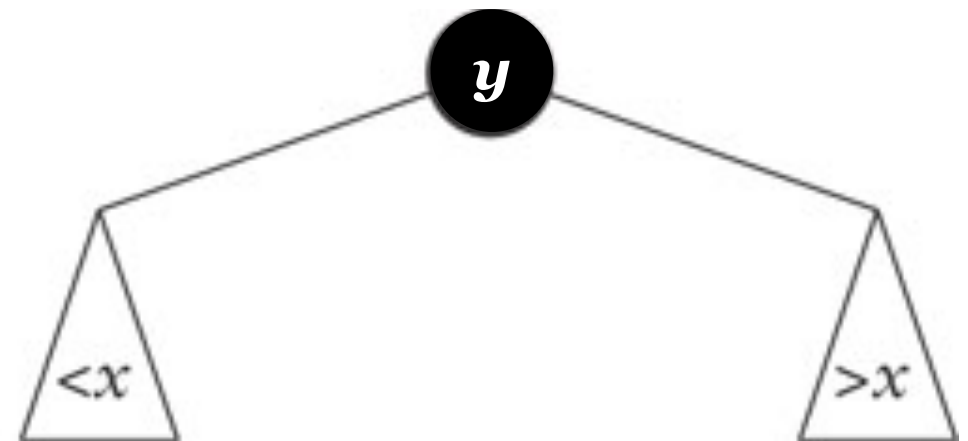
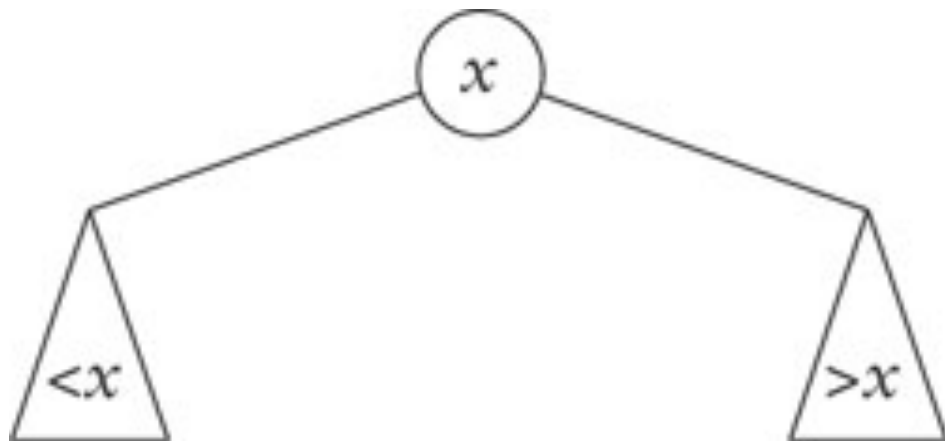
Ett exempel på ett 2-3-4-träd:



2-3-4-träd $\equiv$ rödsvarta träd

Ett rödsvart träd är ekvivalent med ett 2-3-4-träd:

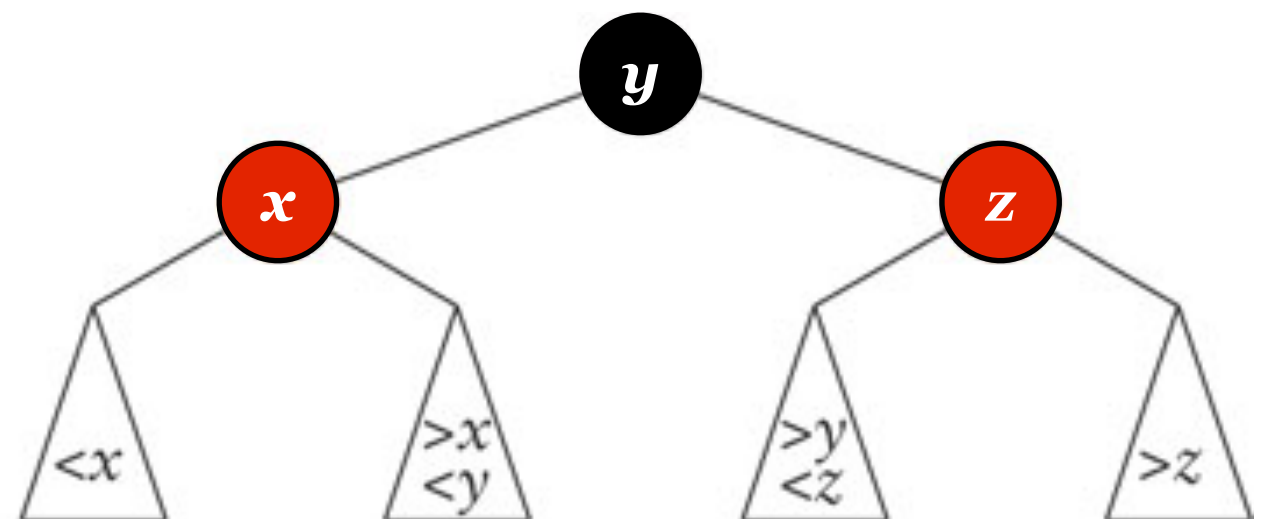
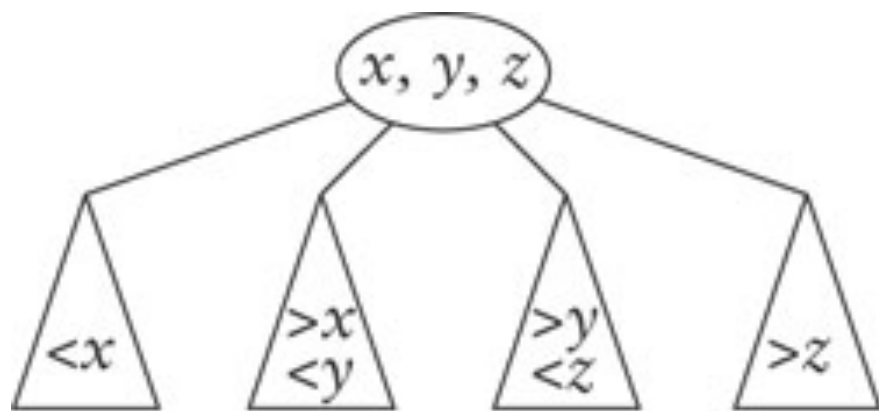
● en 2-nod är en svart nod



2-3-4-träd $\equiv$ rödsvarta träd

Ett rödsvart träd är ekvivalent med ett 2-3-4-träd:

• en 4-nod är en svart nod med två röda barn



2-3-4-träd $\equiv$ rödsvarta träd

Ett rödsvart träd är ekvivalent med ett 2-3-4-träd:

• en 3-nod är en svart nod med ett rött barn

