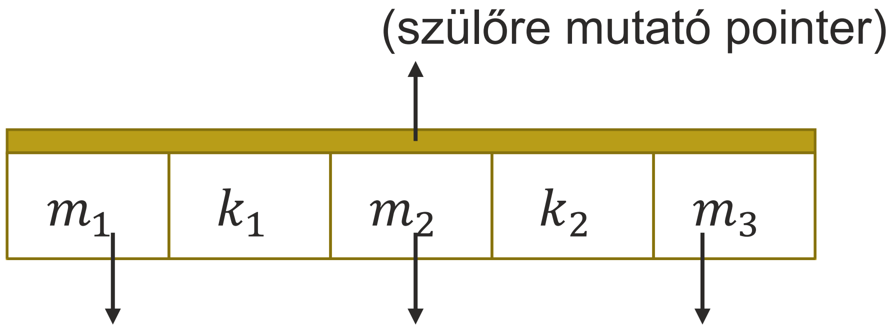
ADATSZERKEZETEK ÉS ALGORITMUSOK

2-3 fák

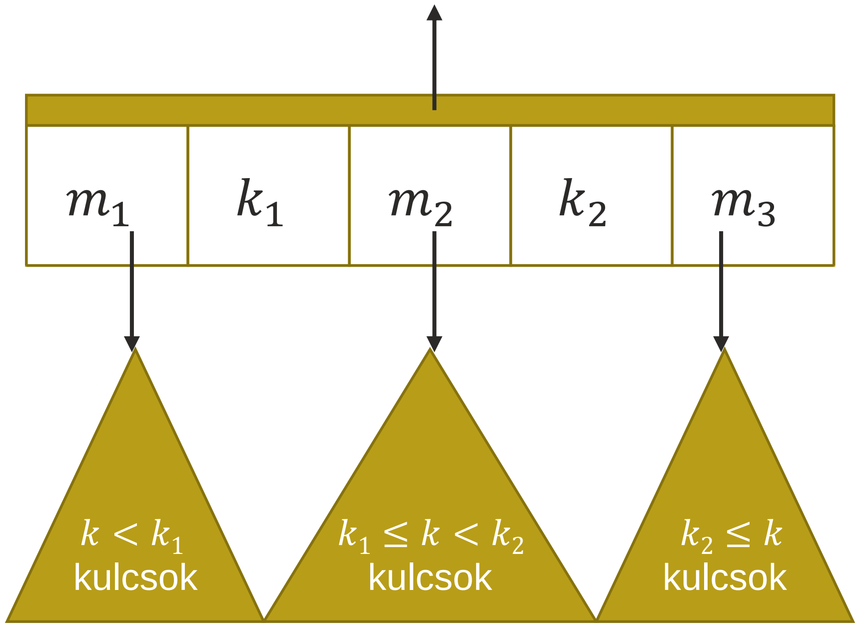
2-3 fák

- Hatékony keresőfa-konstrukció.
- Ez is fa, de a binárisnál bonyolultabb:
  egy nem-levél csúcsnak 2 vagy 3 fia lehet.
- A 2-3-fa egy (lefelé) irányított gyökeres fa, melyre:
  - A rekordok a fa leveleiben helyezkednek el, a kulcs értéke szerint balról jobbra növekvő sorrendben.
  - Egy levél egy rekordot tartalmaz.
  - Minden belső (azaz nem levél) csúcsból 2 vagy 3 él megy lefelé
    - Ennek megfelelően a belső csúcsok egy, illetve két $k \in U$ kulcsot tartalmaznak.
  - A belső pontokban fizikailag ilyen szerkezet van:



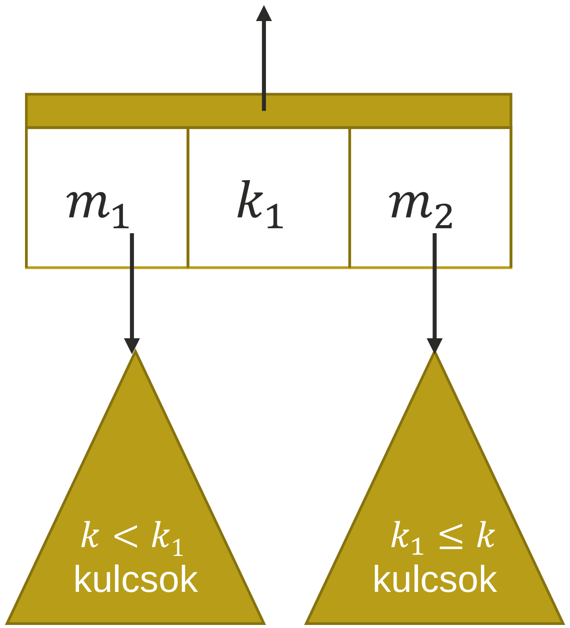
2-3 fák

- Logikailag a belső csúcs kétféle:
  - 3 gyerek
    - Itt m_1; m_2; m_3 mutatók a csúcs részfáira, k_1, k_2 pedig U-beli kulcsok, melyekre $k_1 < k_2$.
    - Az m_1 által mutatott részfa minden kulcsa kisebb, mint k_1.
    - Az m_2 részfájában k_1 a legkisebb kulcs, és minden kulcs kisebb, mint k_2.
    - Végül m_3 részfájában k_2 a legkisebb kulcs.



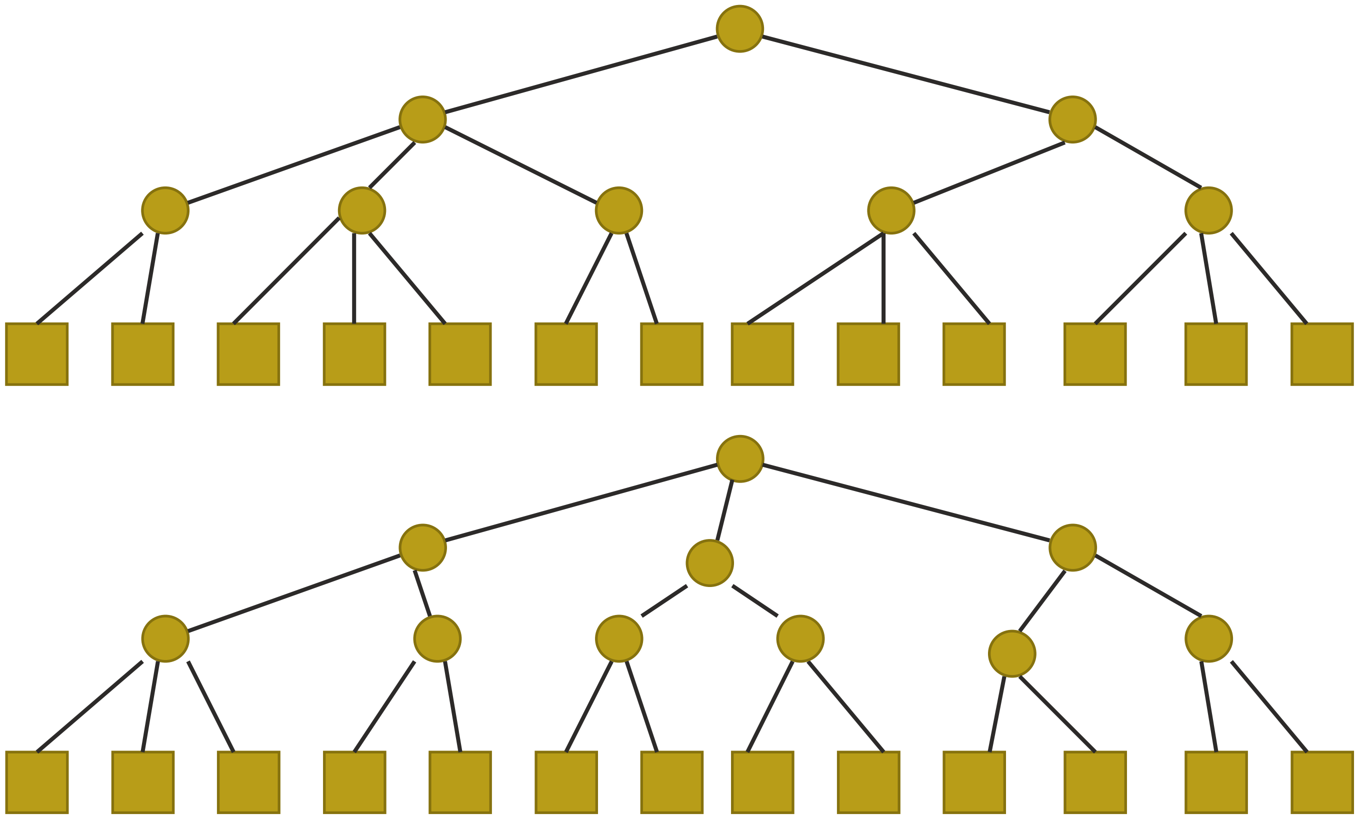
2-3 fák

- Logikailag a belső csúcs kétféle:
  - 2 gyerek
    - Itt m_1, m_2 mutatók a csúcs részfáira, k_1 pedig U-beli kulcs.
    - Az m_1 által mutatott részfa minden kulcsa kisebb, mint k_1.
    - Az m_2 részfájában k_1 a legkisebb kulcs.
    - A két típus közötti váltást csak logikailag kezeljük.



2-3 fák

- Példa két különböző szerkezetű 2-3 fára, $n = 13$-ra



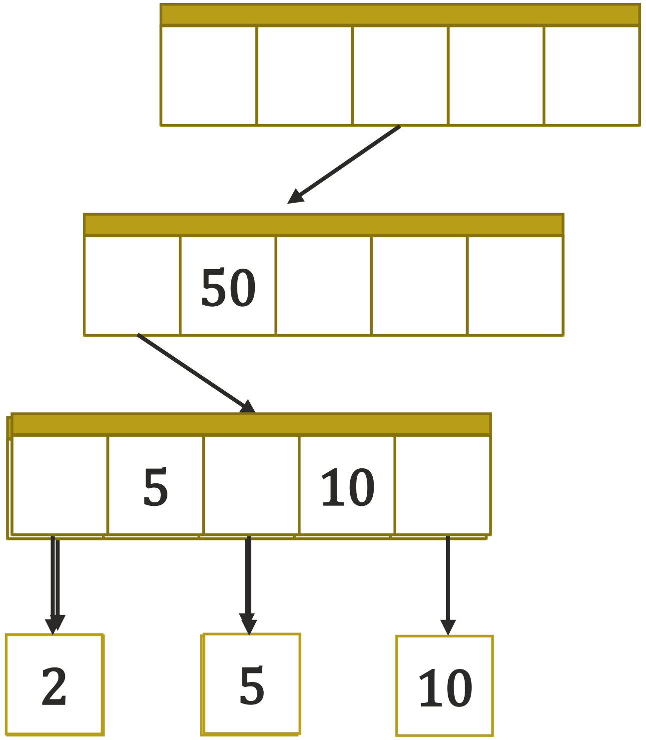
2-3 fák

- Megjegyzés:
  - $n = 0$
    - t=NIL vagy üres gyökér
  - $n = 1$
    - Kivételesen a gyökérnek 1 gyermeke van
- Összefüggés n és h között:
  - $2^h < n < 3^h \Rightarrow h \leq \log_2 n$
  - Még 2-es elágazási tényező esetén is korlátos a fa magassága!

2-3 fák

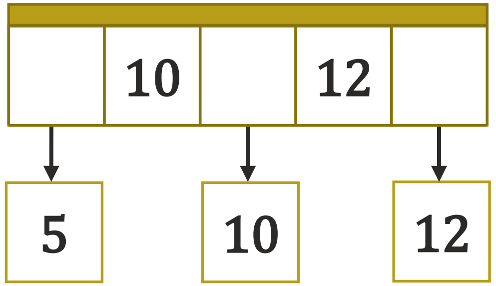
- Műveletek
  - Keresés
    - Összehasonlítások száma
    - $0, 1, \ldots, h-1$ magasságban: 1 vagy 2
    - h magasságban: 1
    - $T(n) < 2h + 1 < 2\log_2 n + 1 = \Theta(\log_2 n)$

2-3 fák – beszúrás

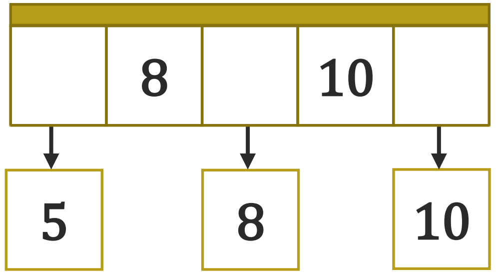


- Kereséssel meghatározzuk a helyét
- I. A legalsó belső pontnak 2 gyereke van (elfér még egy harmadik is)

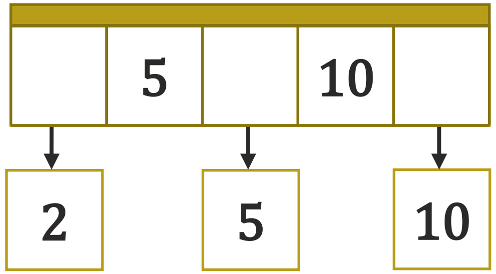
a. $k = 12$



b. $k = 8$

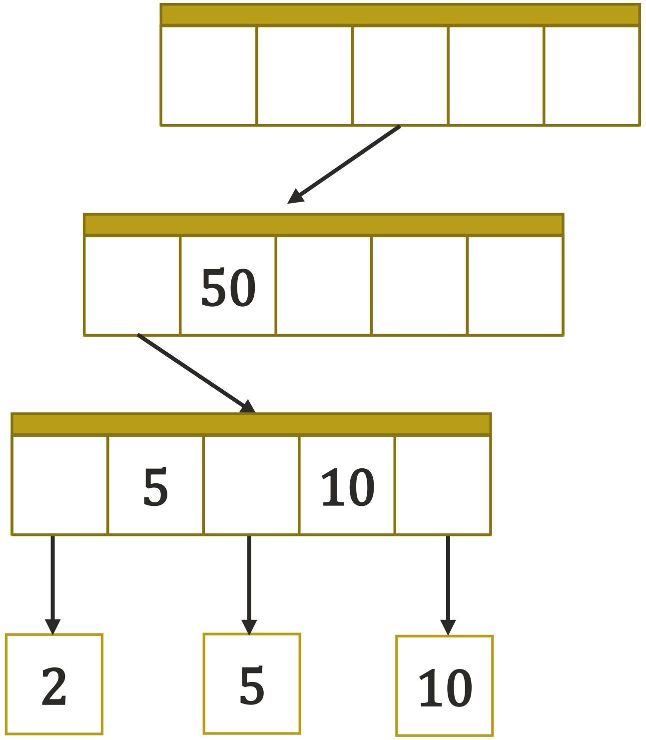


c. $k = 2$

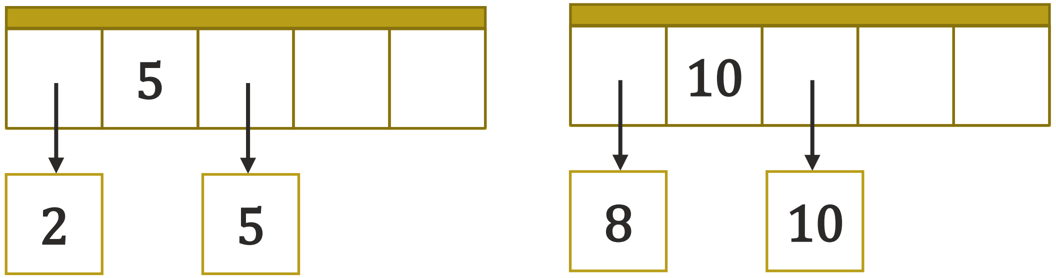


2-3 fák – beszúrás

- Kereséssel meghatározzuk a helyét
- II. A legalsó belső pontnak 3 gyereke van

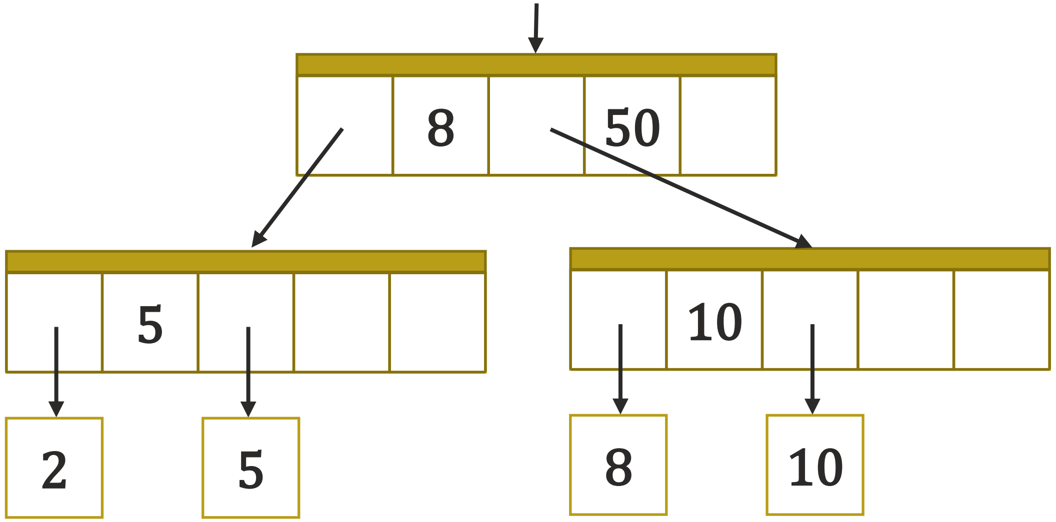


$k = 8$
A megoldás a csúcsvágás



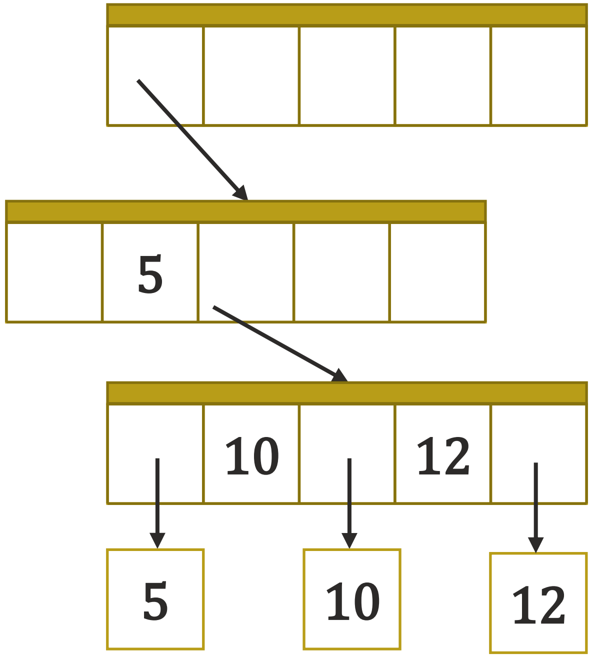
2-3 fák – beszúrás

- II. A legalsó belső pontnak 3 gyereke van. (folyt.)
  - Ha a szülőnek eleve 3 gyereke volt, akkor itt is csúcsvágásra van szükség, és így tovább felfelé. Ha valahol ezen az úton van egy kétgyermekes belső csúcs, akkor ott megáll a beszúrás, mert annak lehet 3 gyereke
  - Ha ezen az úton minden belső pontnak 3 gyereke volt, akkor a csúcsvágás felgyűrűzik a gyökérig. Ekkor a felfelé vágott gyökér fölé egy új gyökeret kell tenni, ami megnöveli a fa magasságát!
  - Az előző példát véve alapul:

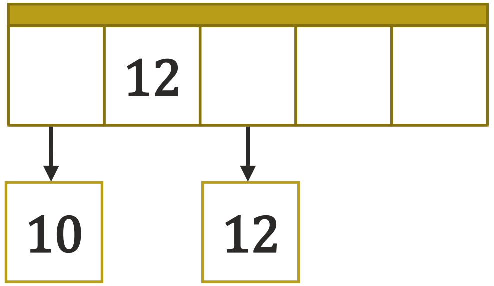


2-3 fák – törlés

- Megkeressük a törlendő kulcsot.
- I. A kulcs szülőjének 3 gyereke van
  - tehát neki 2 testvére van



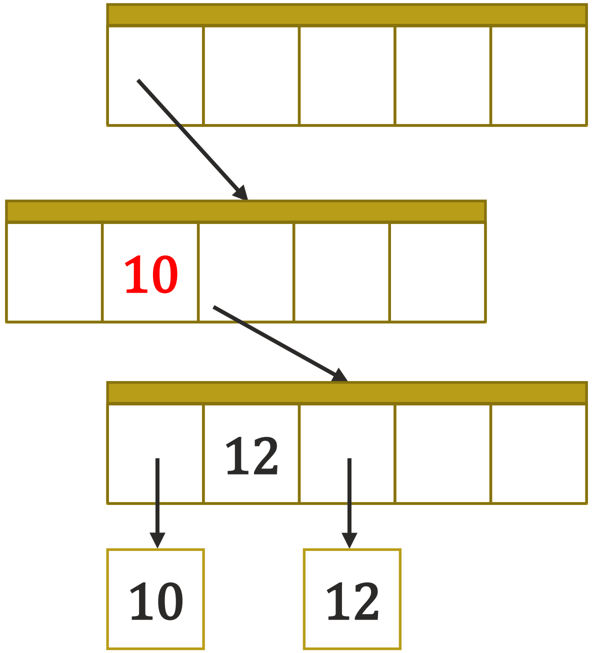
$k = 5$



korrekció a szülőben

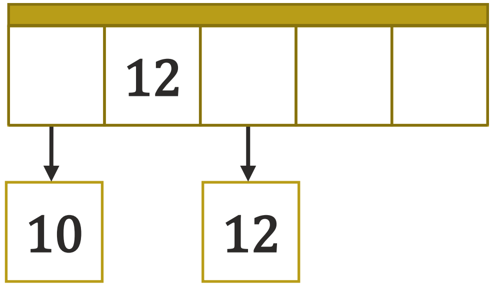
5-ről 10-re

2-3 fák – törlés



- Megkeressük a törlendő kulcsot.
- I. A kulcs szülőjének 3 gyereke van
  - tehát neki 2 testvére van

$k = 5$

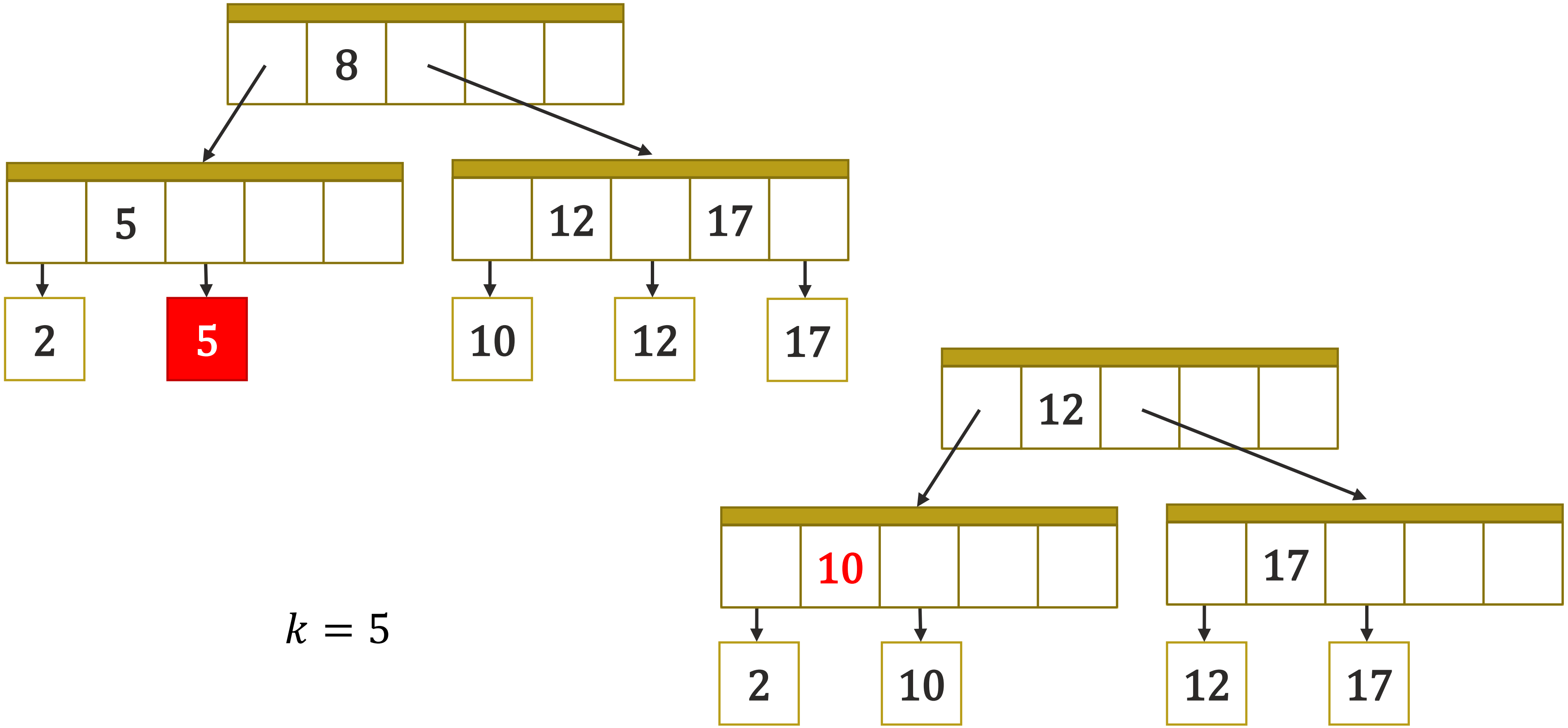


korrekció a szülőben

5-ről 10-re

2-3 fák – törlés

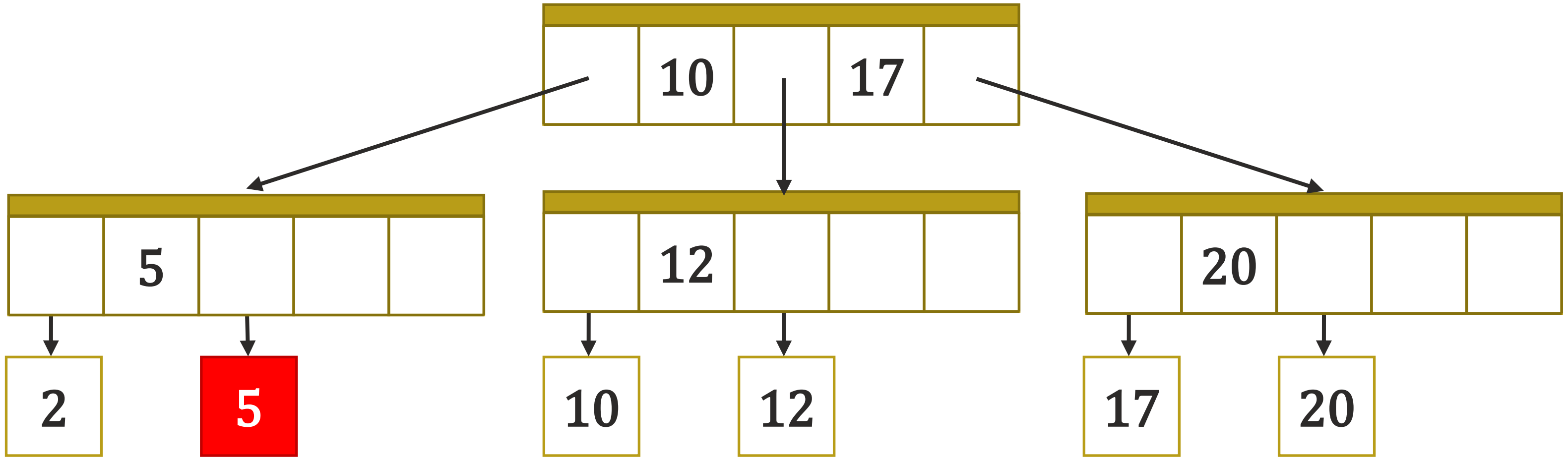
- II. A törlendő elemnek csak 1 testvére van, (a szülőnek 2 gyereke van)
  - II/1. Ha a szülőnek van 3 gyerekes testvére, akkor az 1 gyereket „átad"

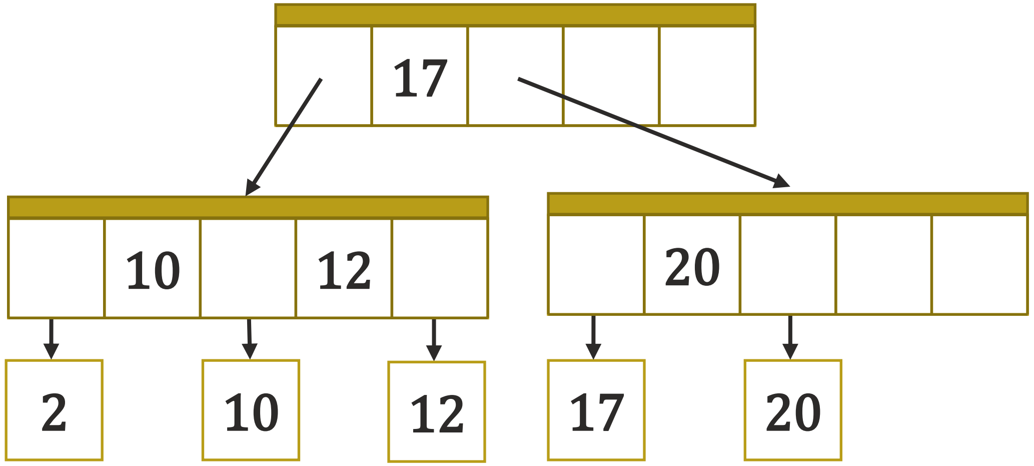


$k = 5$

2-3 fák – törlés

- II. A törlendő elemnek csak 1 testvére van, (a szülőnek 2 gyereke van)
  - II/1. Ha a szülőnek nincs 3 gyerekes testvére, akkor csúcsot vonunk össze

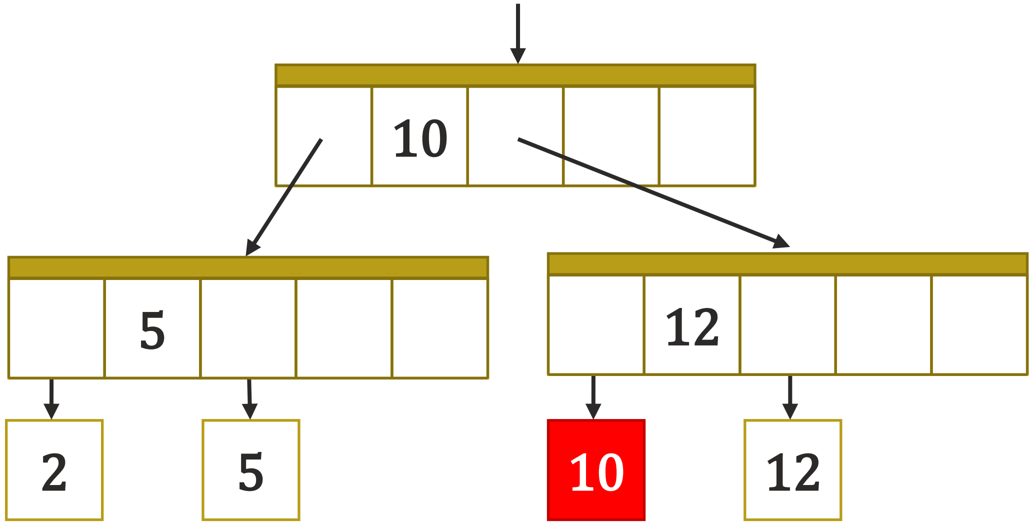




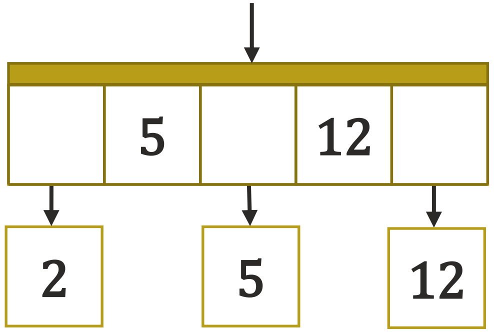
$k = 5$

2-3 fák – törlés

- Szükség esetén a csúcsösszevonásokat folytatjuk felfelé.
  - Ez felgyűrűzhet a gyökérig – így a fa magassága csökkenhet



$k = 10$



2-3 fák

- Műveletek költsége:
  - $T(n) = \mathcal{O}(h) = \mathcal{O}(\log_2 n)$

B-fák

Következő téma