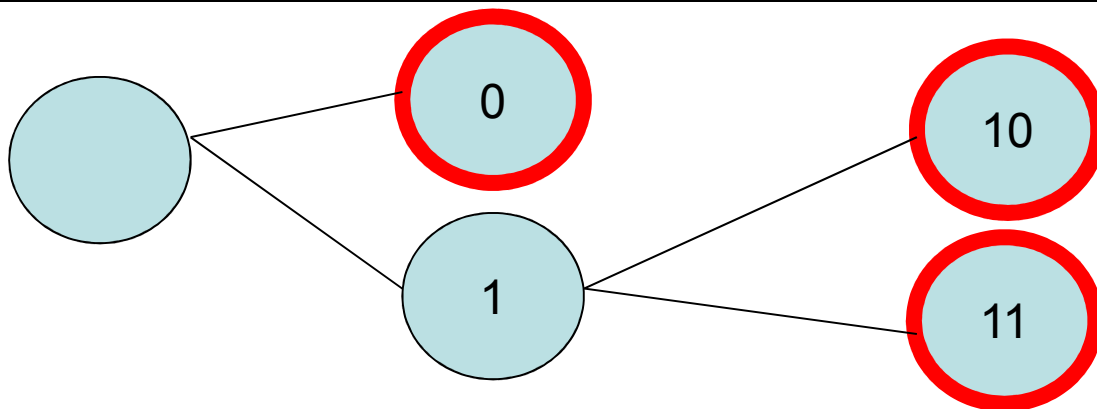
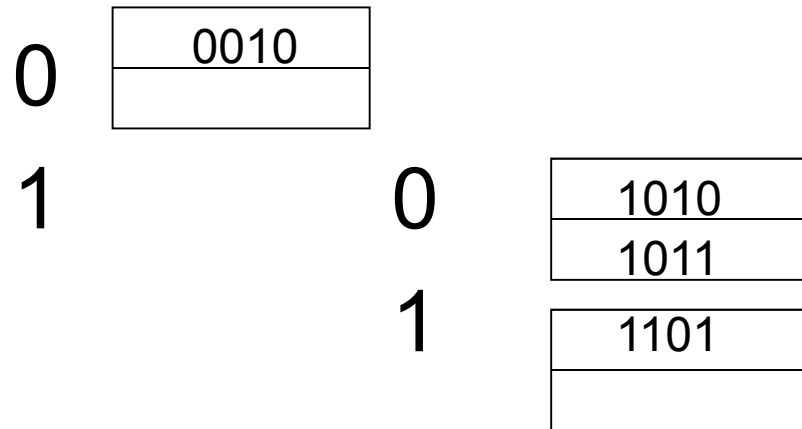
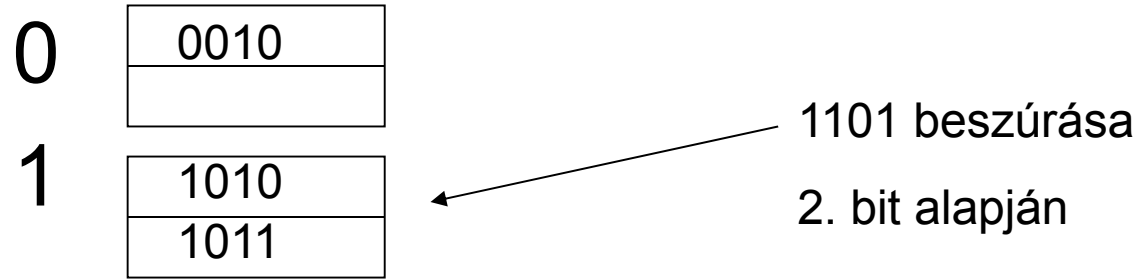


# Indexelés

- **Dinamikus hasító indexek:**
  - **kiterjeszhető (expandable)**
  - **lineáris**
- Előre nem rögzítjük a kosarak számát, a kosarak száma beszúrásakor, törléskor változhat.
- **Kiterjeszhető hasító index:**
- Minden kosár 1 blokkból áll. Keresési költség: **1**.
- Legyen  **$k > \log$  (a rekordok várható számának felső korlátja)**,
  - azaz  $k$  hosszú bináris sorozatból több van, mint ahány rekord
- **A  $h$  hasító függvény értéke egy  $k$  hosszú bináris sorozat.**
- Minden kosárhoz tartozik egy legfeljebb  $k$  hosszú bináris sorozat (kódszó).
- A kosarakhoz rendelt kód **prefix kód**. A maximális kód hossza legyen  $i$ .
- A  $h(K)$   $k$  hosszú kódnak vegyük az  **$i$  hosszú elejét**, és **azt kosarat, amelynek kódja a  $h(K)$  kezdő szelete**. Ha van hely a kosárban, tegyük bele a rekordot, ha nincs, akkor nyissunk egy új kosarat, és a következő bit alapján osszuk ketté a telített kosár rekordjait. Ha ez a bit mindegyikre megegyezik, akkor a következő bitet vesszük a szétosztáshoz, és így tovább.

# Indexelés



# Indexelés

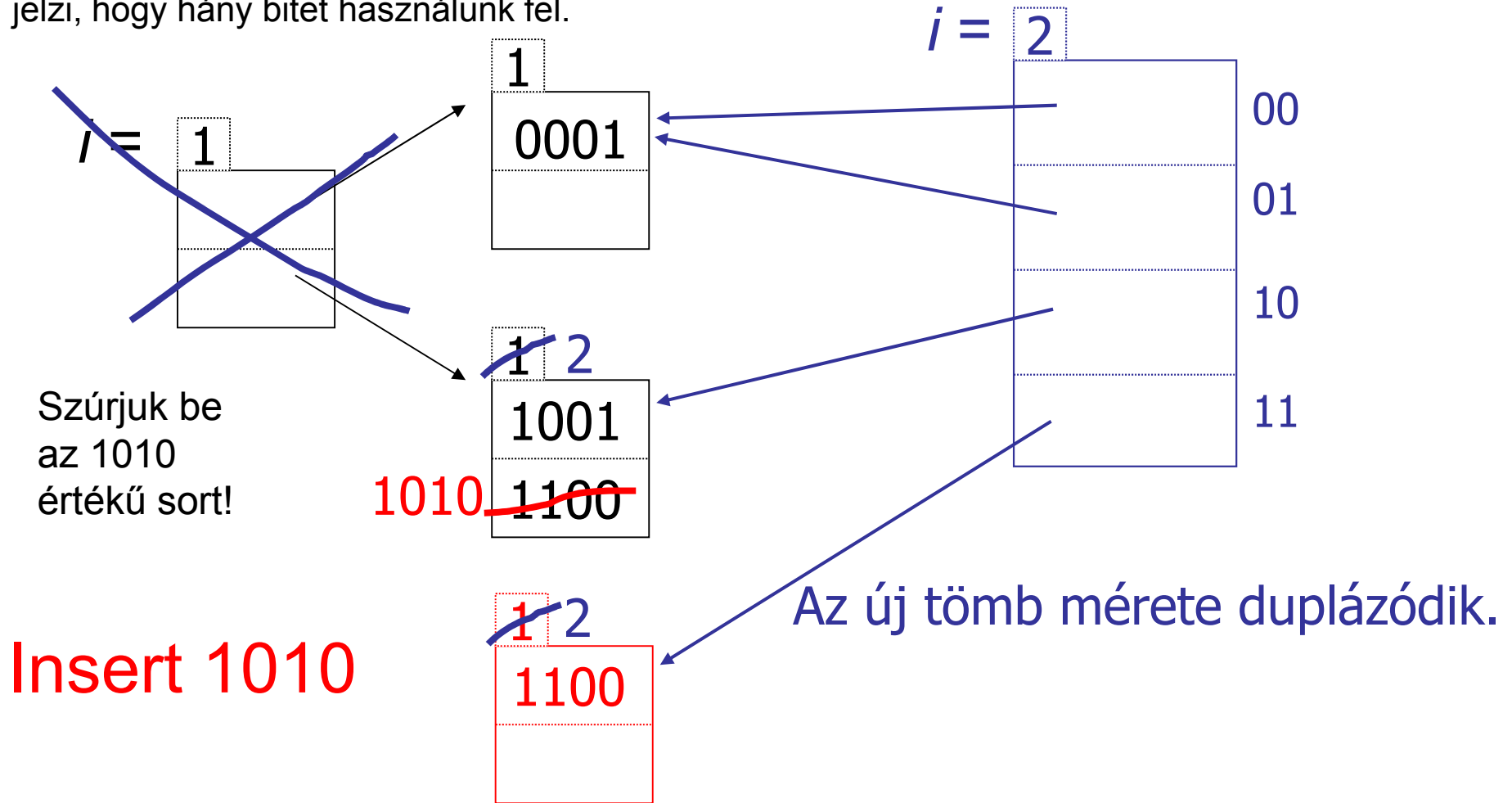
- A bináris fa levelei a kosárblokkok kódszavai. A hasító függvény értékéből annyi bitet használunk, ahányadik szinten szerepel a levél.
- A gráfot a memóriában tárolhatjuk.

**Probléma:** Ha az új sorok hasító értékének eleje sok bitben megegyezik, akkor **hosszú ágak** keletkezhetnek.

**(Nincs kiegyensúlyozva a fa.)**

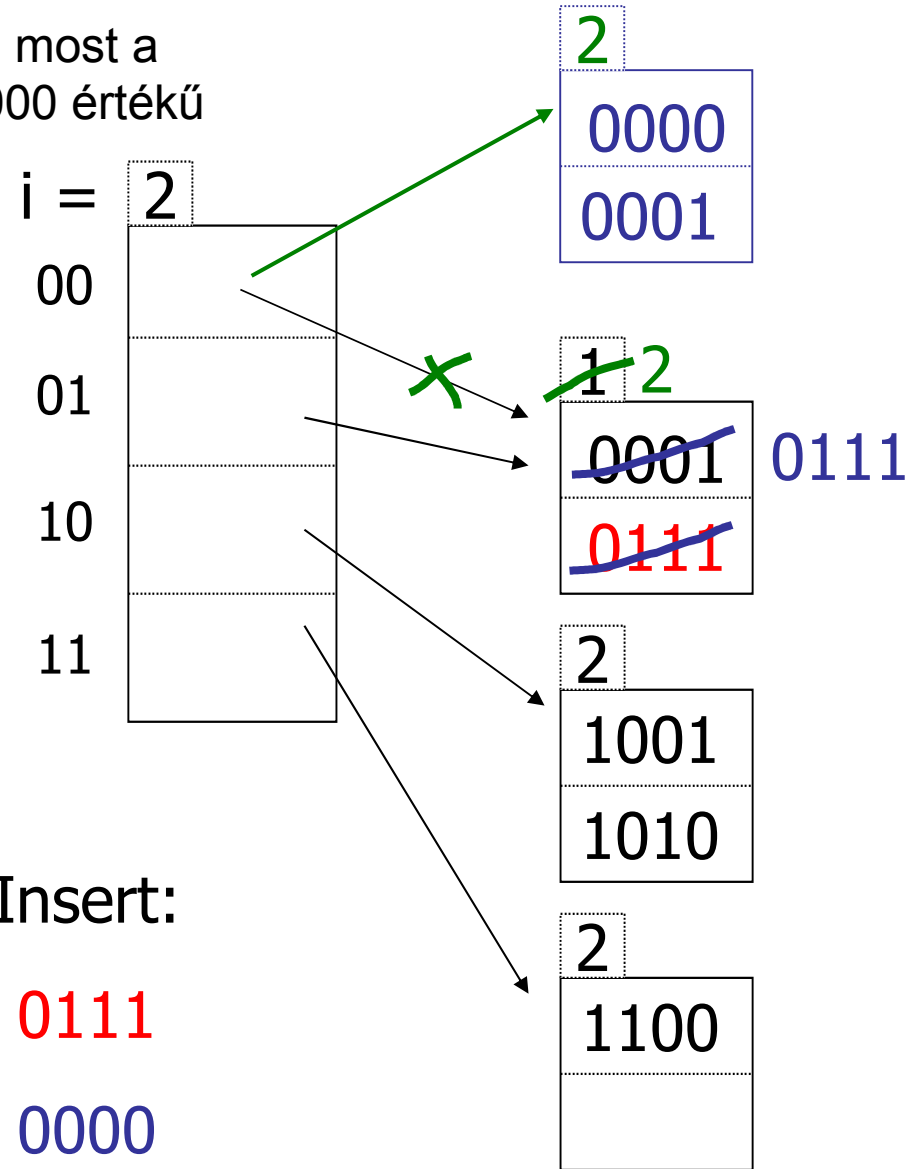
# Indexelés

**A bináris gráfot teljessé is tehetjük.** A gráfot egy tömbbel ábrázolhatjuk. Ekkor minden kosár azonos szinten lesz, de közös blokkjai is lehetnek a kosaraknak. Túlcsordulás esetén **a kosarak száma duplázódik**. Legyen például  $h(k)$  4 bites és 2 rekord férjen egy blokkba. Az  $i$  jelzi, hogy hány bitet használunk fel.

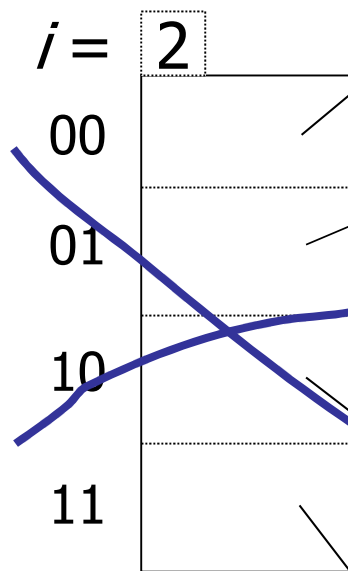


# Indexelés

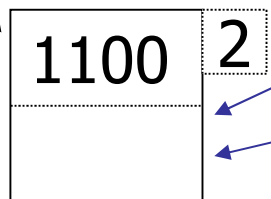
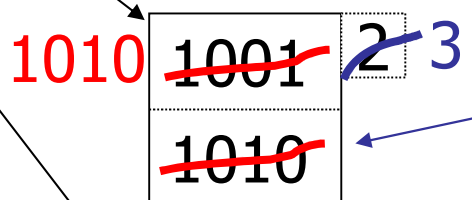
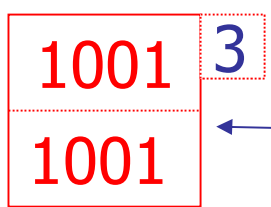
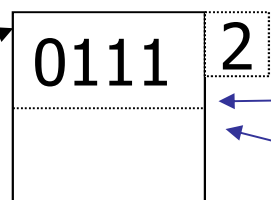
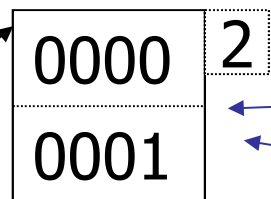
Szűrjük be most a  
0111 és 0000 értékű  
sorokat!



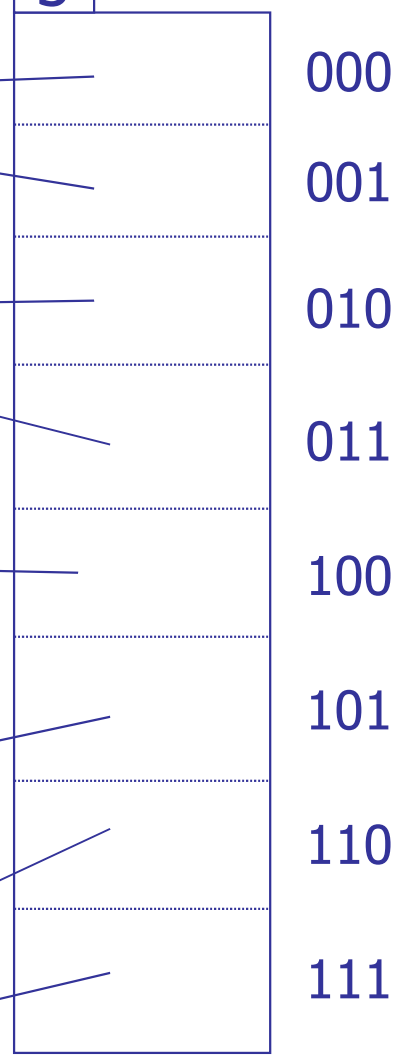
Szűrjük be az  
1001 sort!



### Indexelés



$i = 3$



Insert:

1001