**B-fa**

**Építés:**

-> beszúrúnk az alapba ameddig fér

-> ha túlcsordul akkor beillesztjük a sorba és úgy vágjuk ketté hogy ha ugyanannyi legyen mindkét oldalon ÉS fentre rakunk pointert, ami a másodikra mutat

-> ha a legfelső csordul túl akkor is ugyanaz DE nem kell pointer mutasson a második sorba elég a harmadik sorba a kisebb

**Bitmap**

**Index készítés:**

A table with numbers and numbers

Description automatically generated

-> ahol igaz az állitás 1 kerül, ahol hamis 0

**Lekérdezés:**

A graph of numbers and a number of numbers

Description automatically generated with medium confidence

-> a logikai szabályok érvényesek

**Tömörítés:**

-> megszámoljuk hány db 0-ás van az 1-ig

-> a darabszámot átalakítjuk binárissá

-> a bináris szám számjegyei legyenek n >> unárisan felírva ez n-1 db 1-es és mögé egy 0

-> ez a szakasz átkodolva: unáris + bináris

A number and numbers on a white background

Description automatically generated

-> repeat

**Visszafejtés:**

-> megszámoljuk hány db 1-es van a 0-ig

-> leolvasunk a fenti szám + 1 db számjegyet és az lesz a bináris változat

-> a bináris számot visszafejtve megkapjuk hány db 0-ás van a bitmap elején

-> a 0-ások után kerül egy 1-es

A black text on a white background

Description automatically generated

-> repeat

!! ha 0db 1-es van az pont 0 tehát 1-et kell leolvasni a és azt kell bitből visszaalakítani AKA:

00 = 0 és 01 = 1 db 0-ást jelent

**Hashelés**

-> i = kosárindexhez használt bitek száma, n = kosarak száma

-> vesszük az indexeket és megnézzük az utolsó i számjegyet

-> az alapján betesszük az indexeket a kosarakba >> r+=1

-> ha r/n nagyobb mint a határérték akkor beszúrúnk még egy kosarat, ilyenkor az i száma változhat

!! beszúrásnál ha az adott rekordhoz nincs kosárindex, akkor nézett számjegyekből a legelső eltérhet

!! lehet túlcsordulás blokk ha az r/n nem lépi át a határértéket

