# Számítógépes képelemzés
# 5. előadás

Dr. Balázs Péter

SZTE, Képfeldolgozás és
Számítógépes Grafika Tanszék

# Régió alapú szegmentálás

- Valamilyen szempontból hasonló mintákat, pontokat tartalmazó területek kialakítása
  - Küszöbölés
  - Régió növelés
  - Régió egyesítés
  - Régió szétválasztás
  - Szétválasztás és egyesítés

# Régió növelés

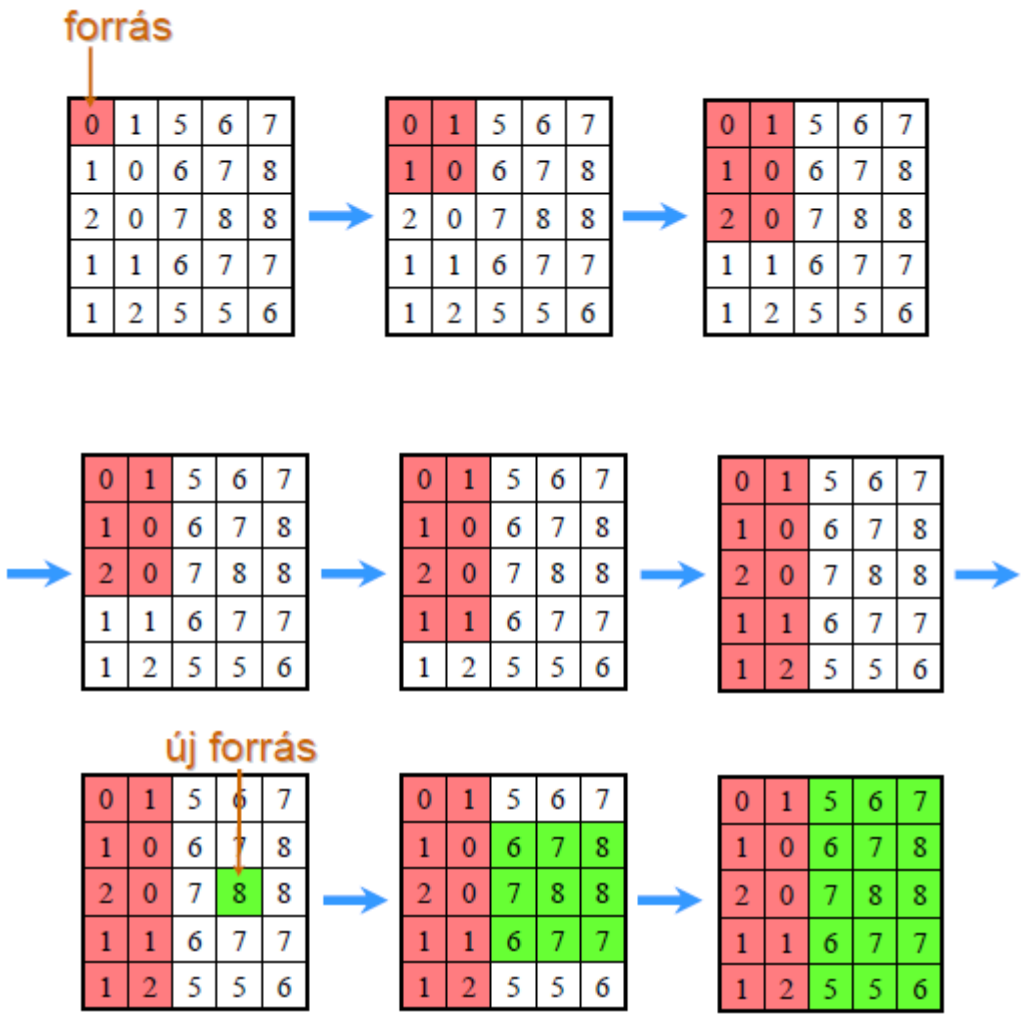
- Válasszuk ki a kép tetszőleges még szegmentálatlan pontját vagy összetartozó pontjait
- Az aktuális régióhoz vegyük hozzá az azzal határos pontok közül a hasonlósági kritériumot kielégítőket
- Ha a régió már nem növelhető és nincs több szegmentálatlan pont, akkor vége, egyébként 1. lépés

# Szomszédság

- 4- és 8-szomszédság
- 4- és 8-összefüggőség
- Jordan-görbe tétel: tetszőleges egyszerű síkgörbe a síkot két részre (külső és belső) osztja
- Összefüggőségi paradoxon

# Példa régiónövelésre

- f(u)-f(v)<3
- 8-szomszédság

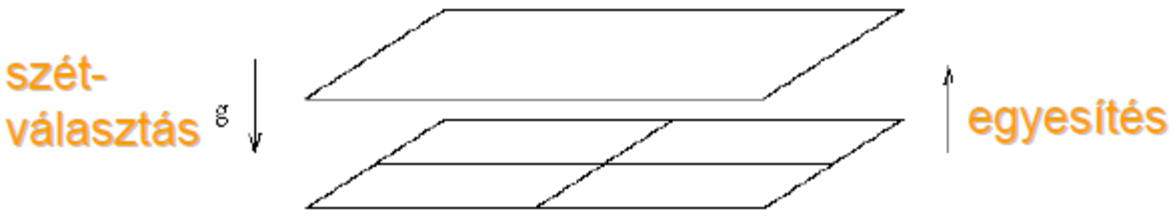


# Régió egyesítés

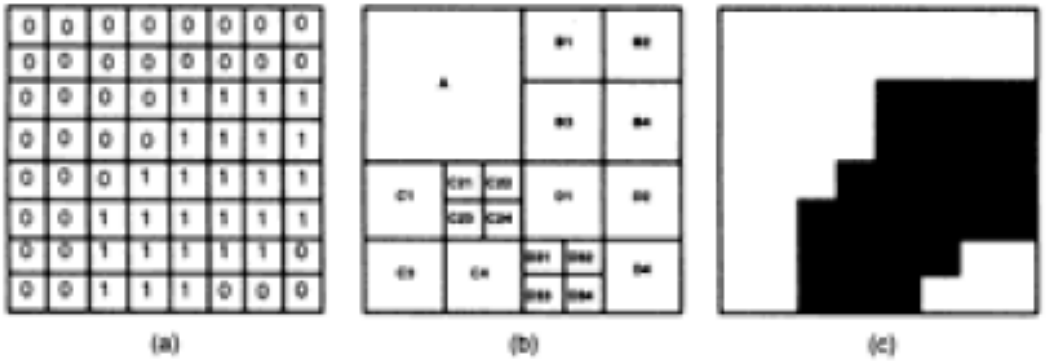
- Cél: túlszegmentálás csökkentése
- Definiáljunk egy kritériumot, ami alapján a szomszédos régiók egyesíthetők
- Egyesítsük a kritériumot kielégítő régiókat
- Pl.: a két régió átlagos szürkeintenzitása megközelítőleg ugyanaz
- Kis régiókat addig csatolunk egy hasonlósági kritérium alapján a szomszédokhoz, amíg el nem tűnnek egy adott méretnél kisebb régiók

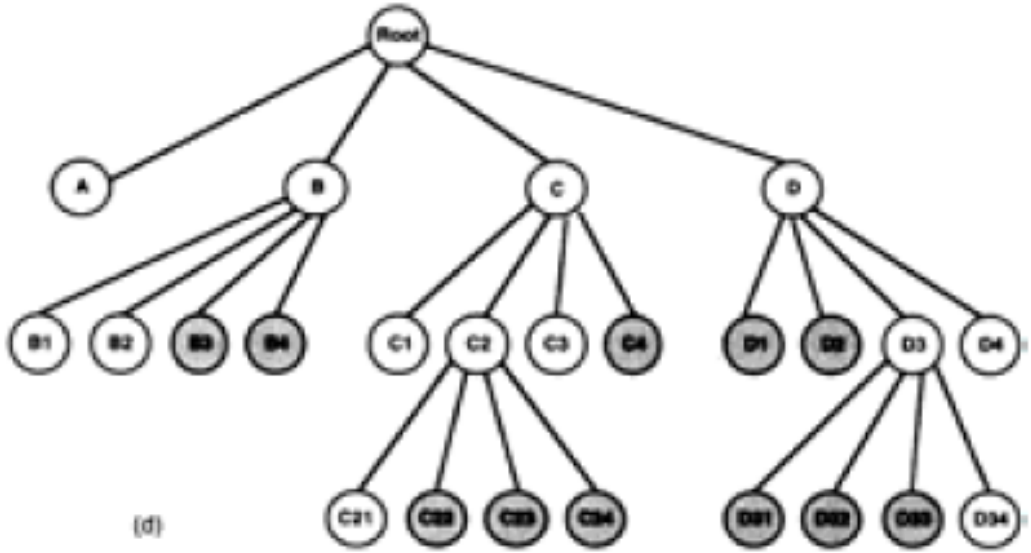
# Régió szétválasztás (és egyesítés)

- Ha az aktuális régiók nem eléggé homogének
- Szétválasztás és egyesítés: nem-uniform régiókat negyedekre bontjuk az uniform és szomszédos régiókat egyesítjük



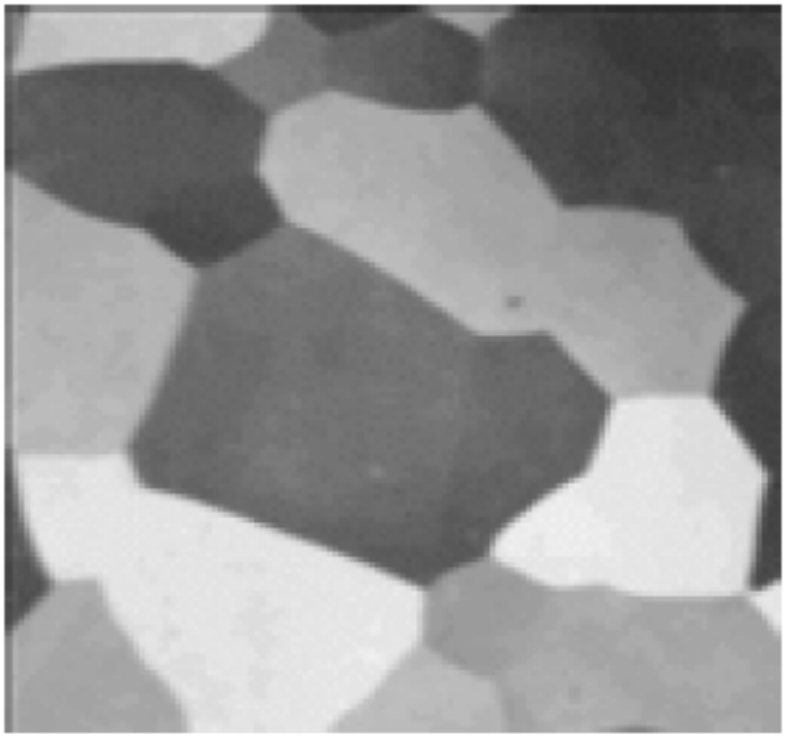
# Quad-tree (4-fa)



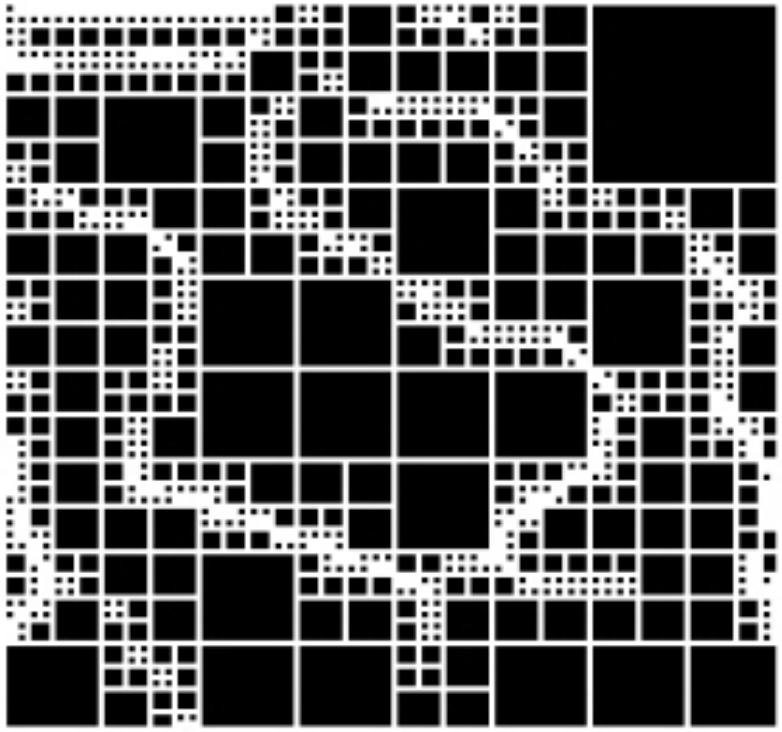




(a) eredeti kép
(b) a szétválasztás eredménye
(c) a kapott két régió
(d) 4-fa (*quad-tree*) reprezentáció

# Egy másik példa



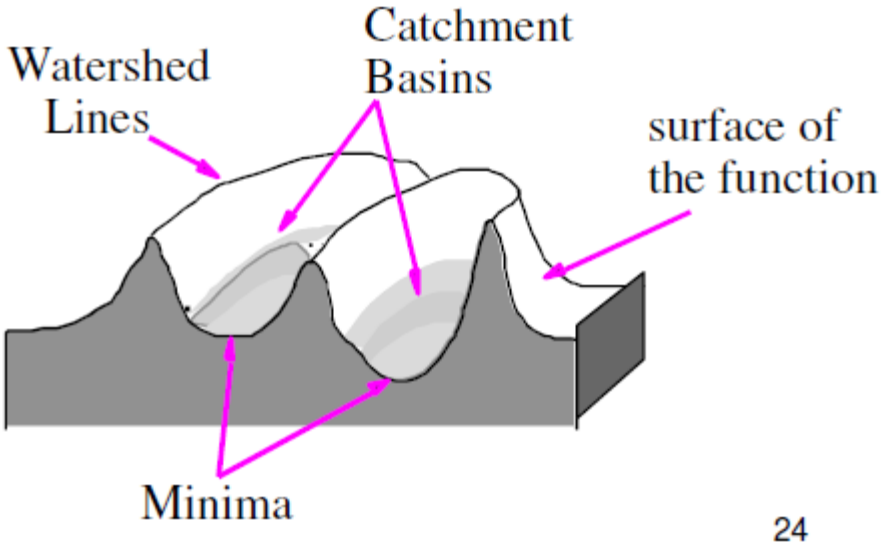
eredeti kép



dekompozíció

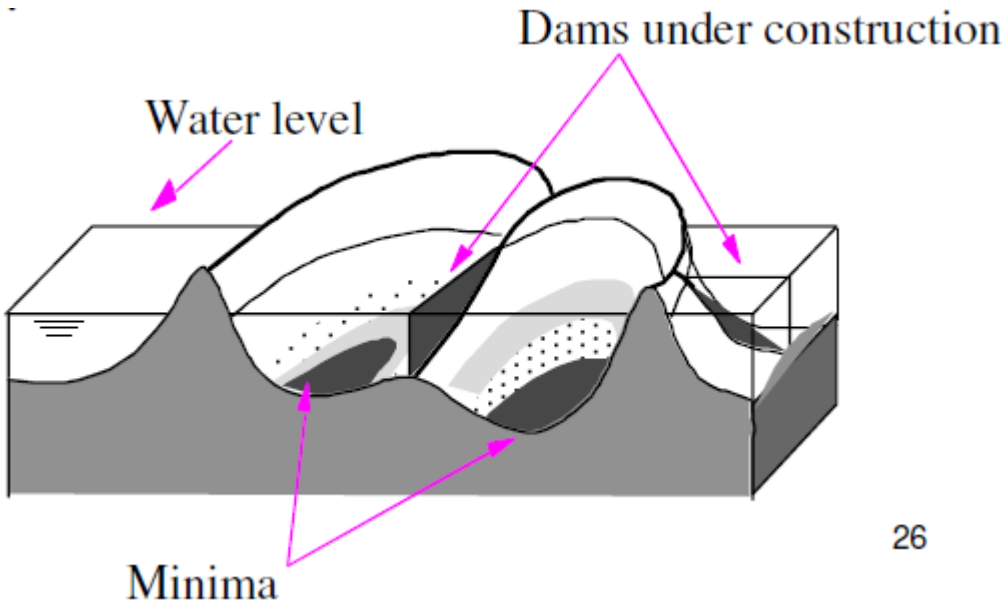
# Szegmentálás vízválasztókkal

- A szürkeárnyalatos képet felületként (domborzatként) fogjuk fel
- Határozzuk meg a régióminimumokat
- Vízgyűjtő medence: pontok, melyről a víz egy határozott régióminimumba folyik
- Vízválasztók: pontok melyekből a víz több irányba folyhat



# A szegmentálás folyamata

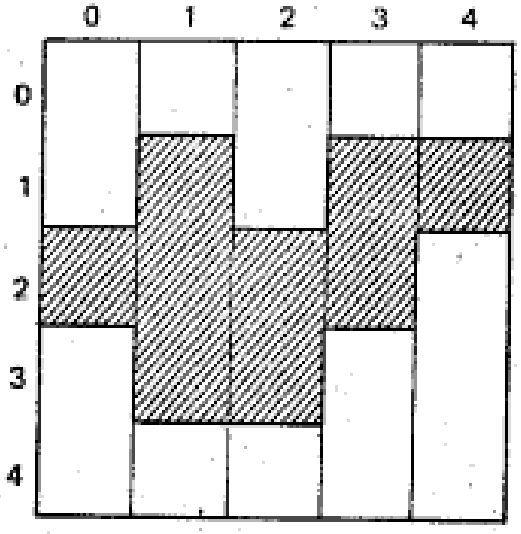
- Régióminimumokat kezdjük el feltölteni vízzel
- Ha diszjunkt vízgyűjtők érintkeznének, emeljünk gátat
- A végén csak a gátak teteje látszik
- Általában túlszegmentáláshoz vezet a sok lokális minimum miatt



# ImageJ megvalósítás

- Plugin letöltése
- Plugins → Filters → Watershed algorithm

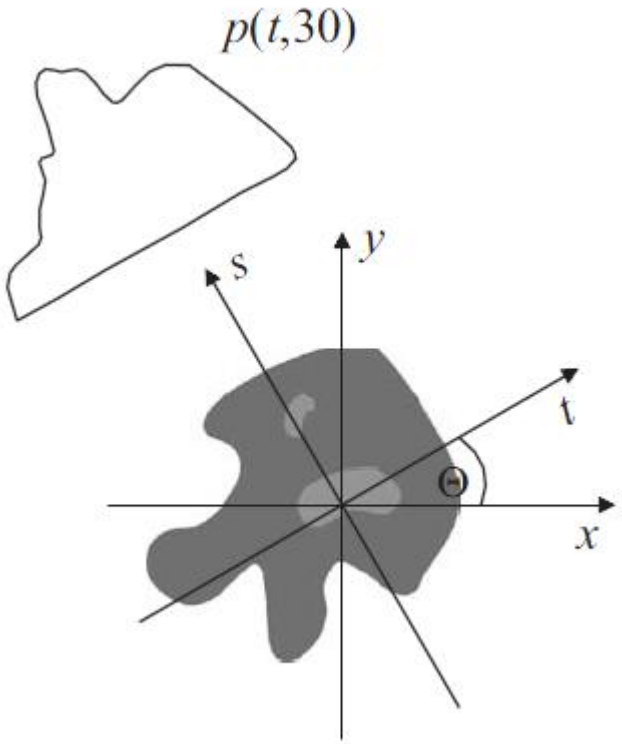
# Futamhossz-kódolás

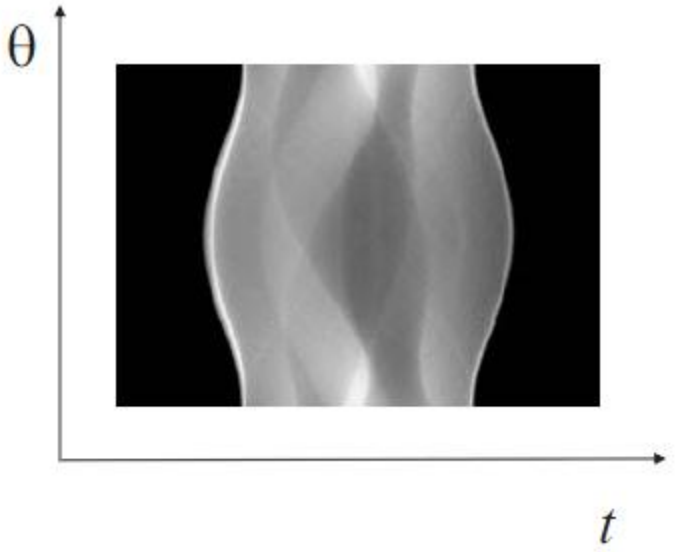



(a) Binary image

| Run-length code | Run # |
|---|---|
| (1, 1)1, (1, 3)2 | 1, 2 |
| (2, 0)4 | 3 |
| (3, 1)2 | 4 |
| . | . |
| . | . |

# Vetületek (Radon Plugin)



# Rekonstrukció

Ryser, 1957 – az *R* sor- és *S* oszlopösszegekből

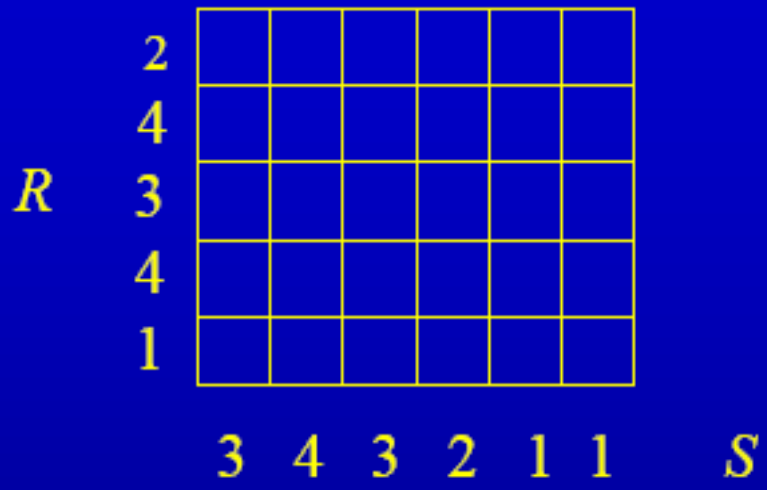
Képezzük *S* nemnövekvő átrendezését $\pi$ permutációval $\rightarrow$ *S*'

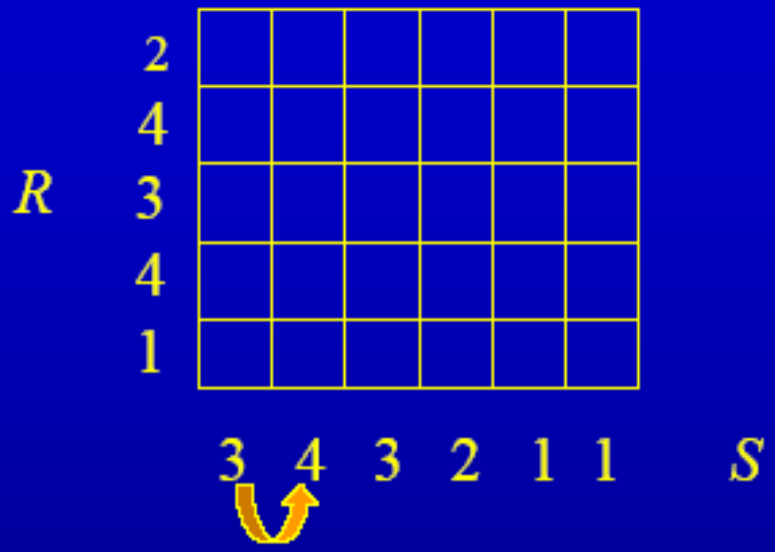
Töltsük fel az oszlopokat balról-jobbra $\rightarrow$ *B* (kanonikus mátrix)

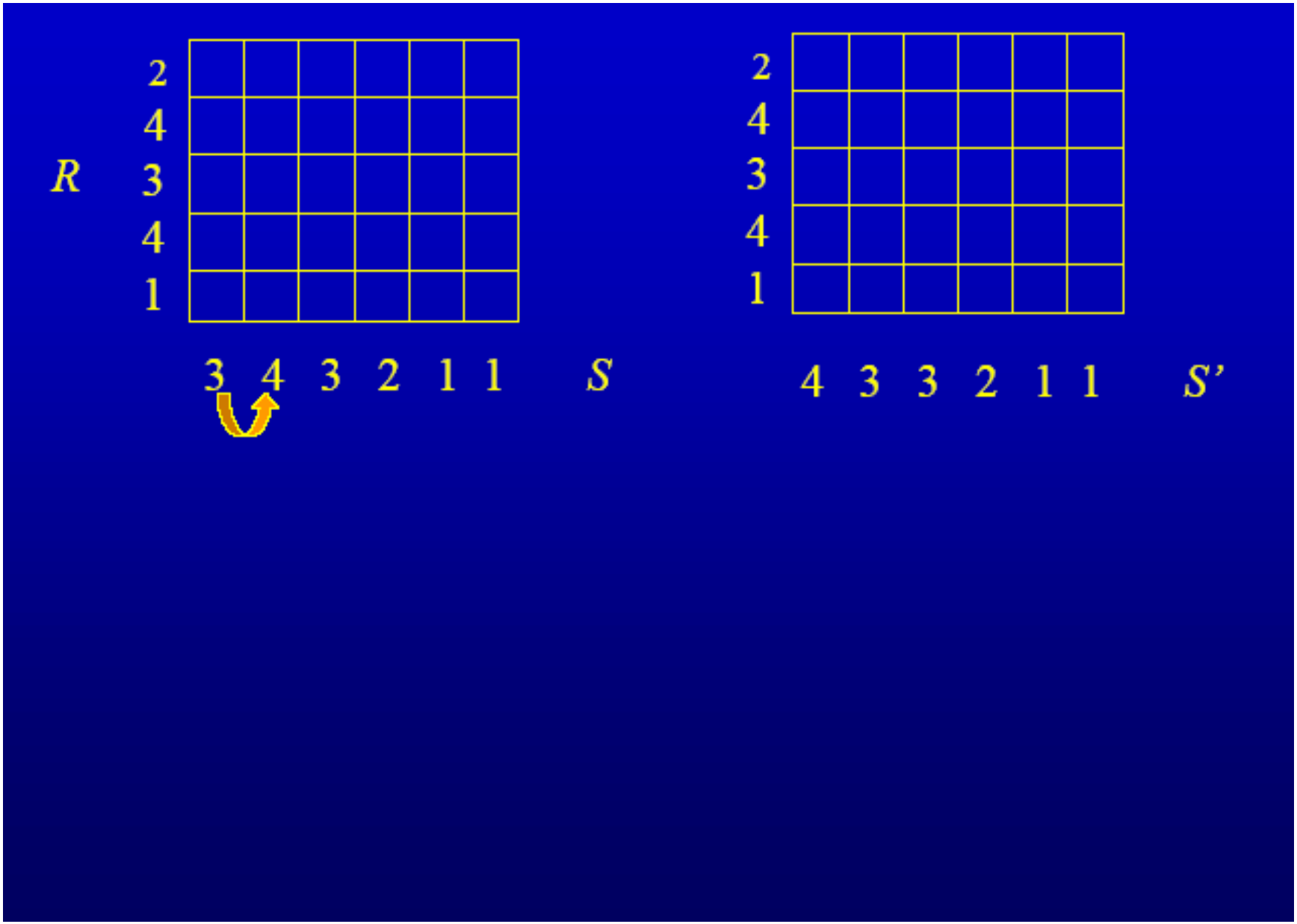
*B* legjobboldali oszlopaiból toljunk el elemeket oda, ahol *S*(*B*) < *S*'

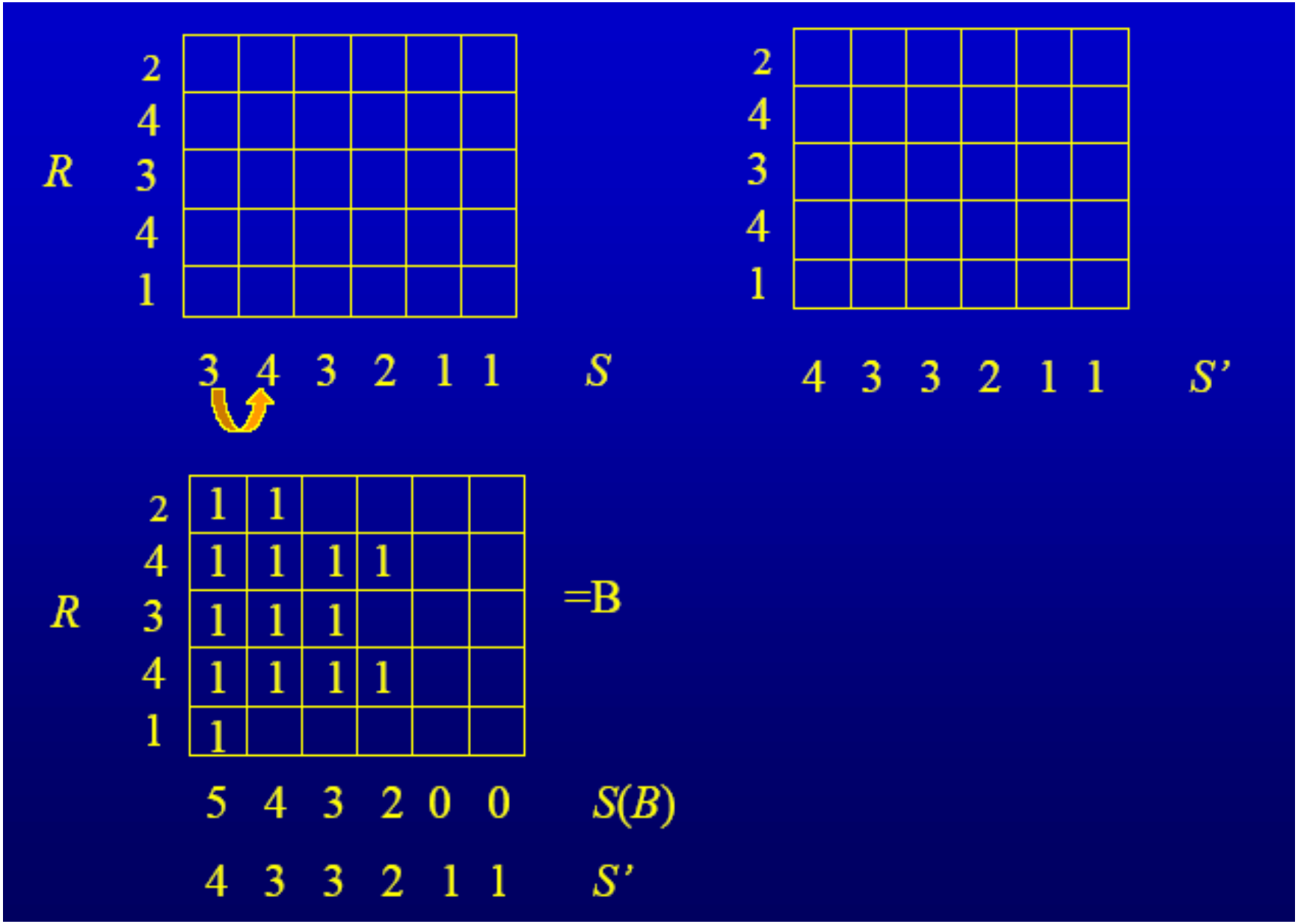
$\pi$ inverzének segítségével rendezzük vissza az oszlopokat

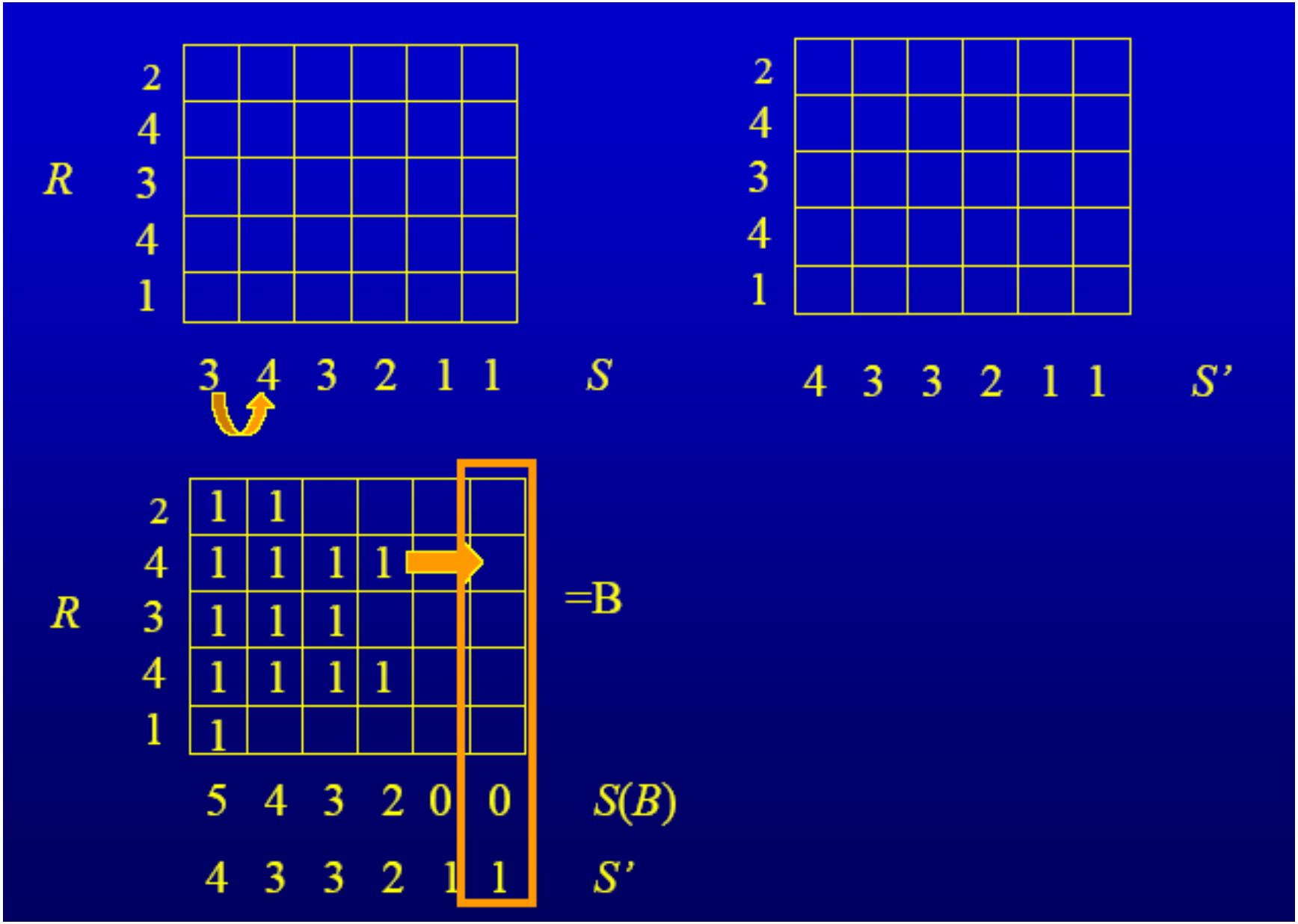
Komplexitás: $O(nm + n\log n)$

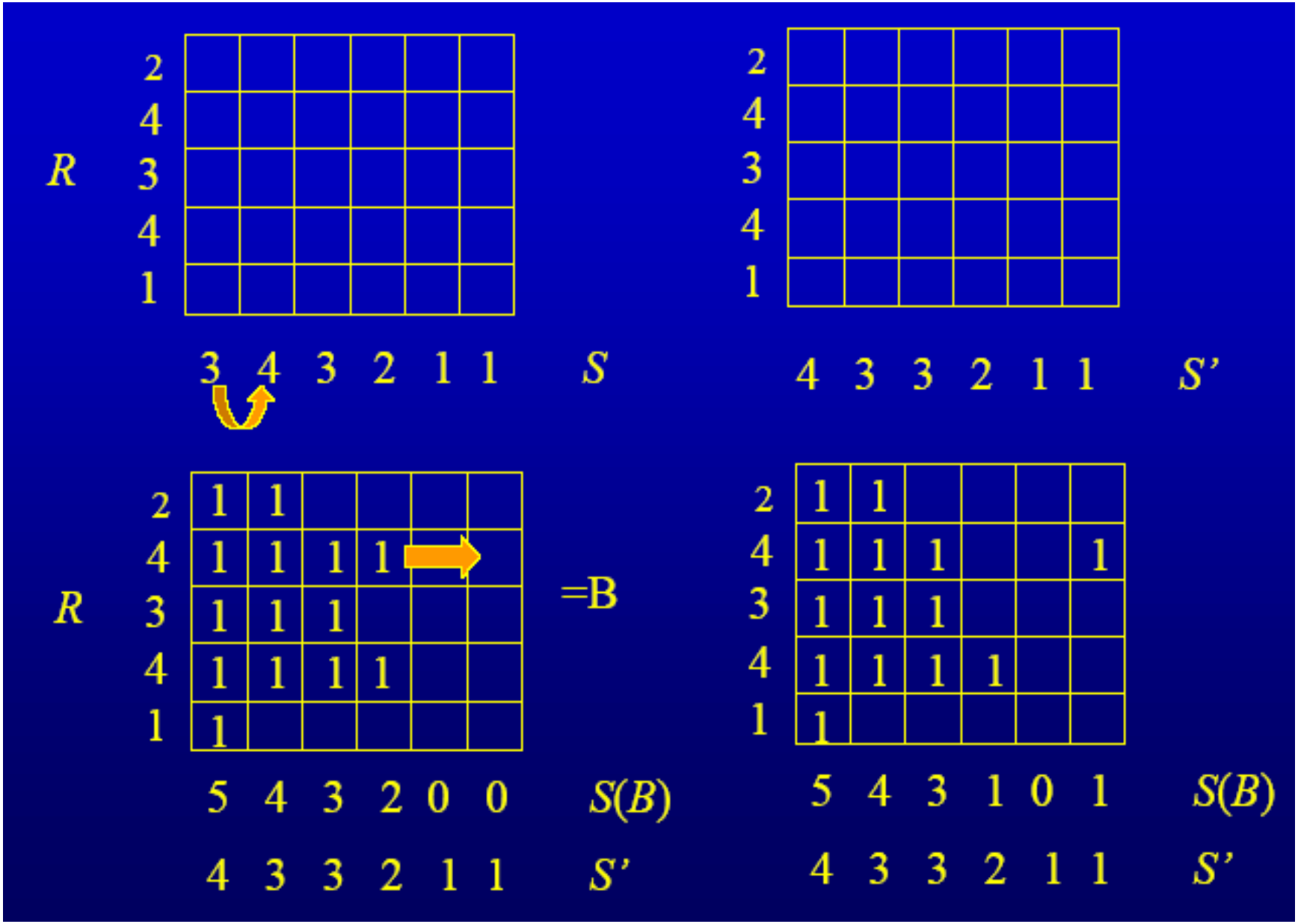

R
2
4
3
4
1
3 4 3 2 1 1
S

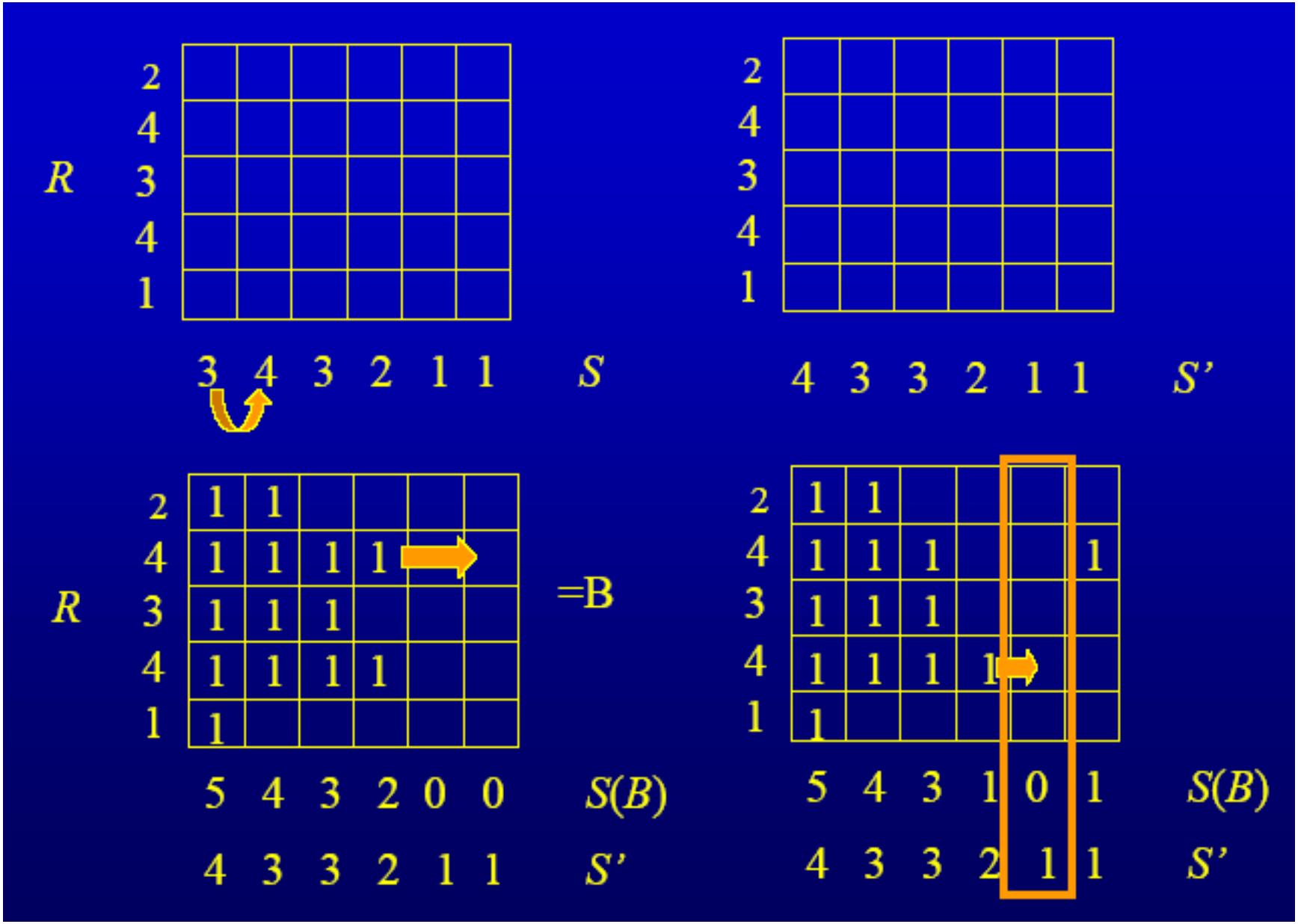

R
2
4
3
4
1
3 4 3 2 1 1
S

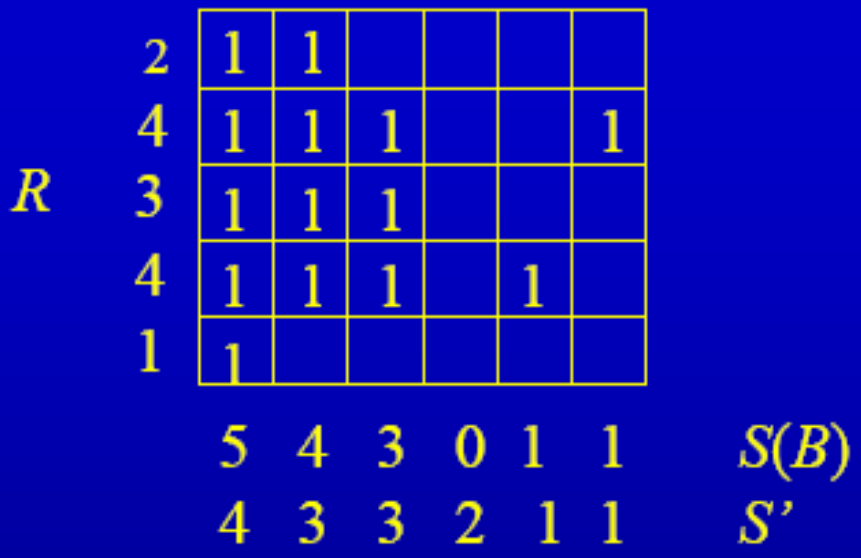

R
2
4
3
4
1
3 4 3 2 1 1
S
2
4
3
4
1
4 3 3 2 1 1
S'

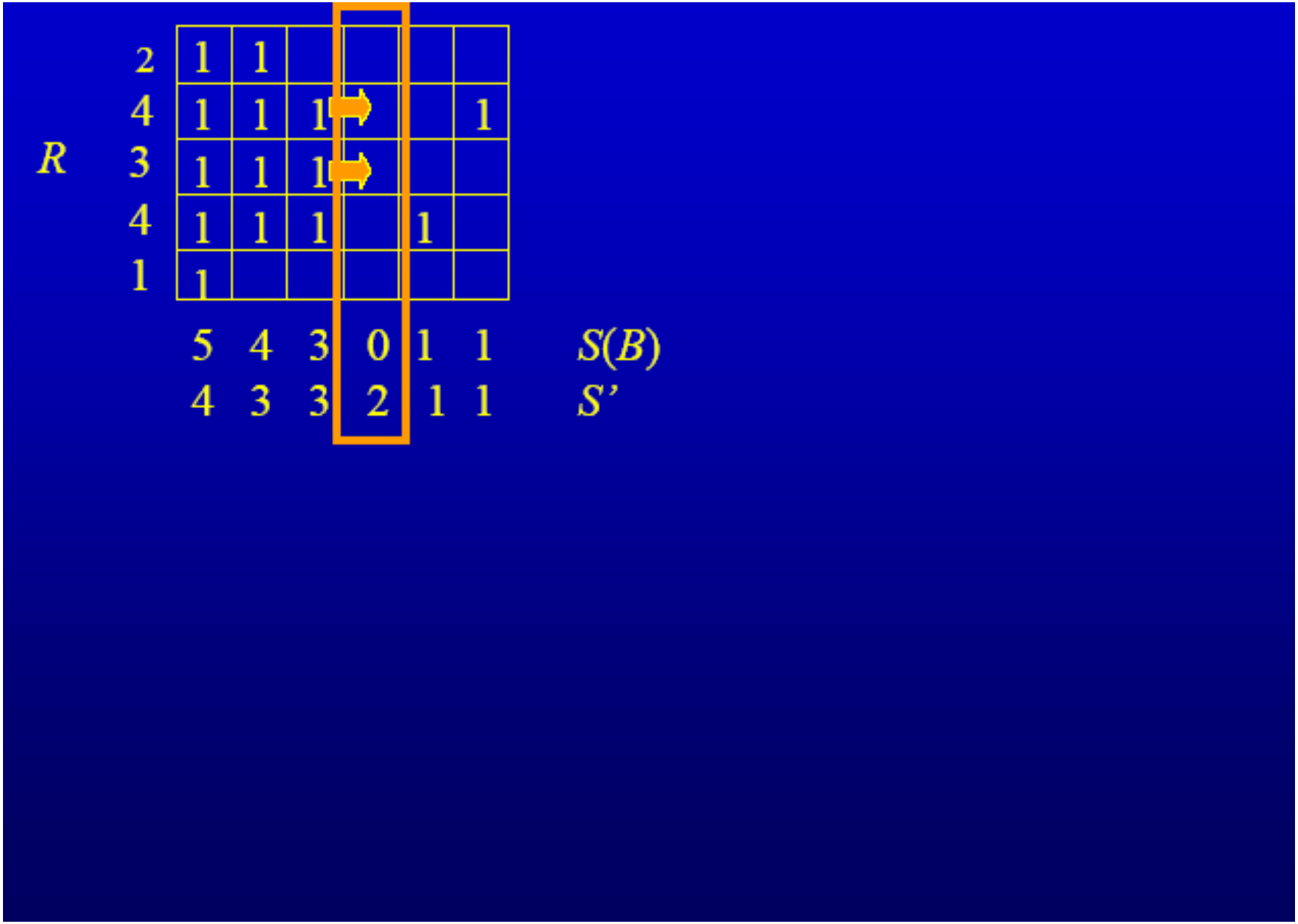

R
2
4
3
4
1
3 4 3 2 1 1
S
2
4
3
4
1
4 3 3 2 1 1
S'
R
2
4
3
4
1
1 1
1 1 1 1
1 1 1
1 1 1 1
1
=B
5 4 3 2 0 0
S(B)
4 3 3 2 1 1
S'

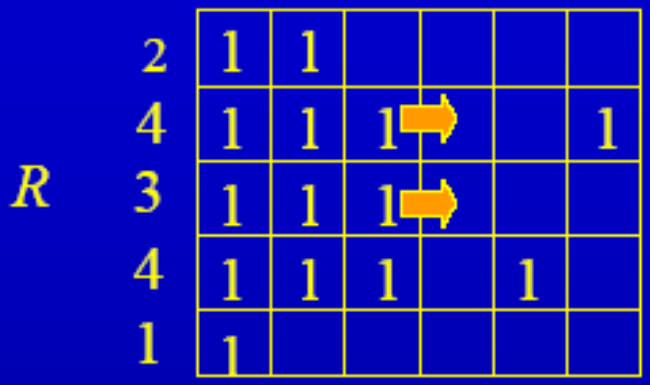

R
2
4
3
4
1
3 4 3 2 1 1
S
2
4
3
4
1
4 3 3 2 1 1
S'
R
2
4
3
4
1
1 1
1 1 1 1
1 1 1
1 1 1 1
1
=B
5 4 3 2 0 0
S(B)
4 3 3 2 1 1
S'


R
2
4
3
4
1
3 4 3 2 1 1
S
2
4
3
4
1
4 3 3 2 1 1
S'
R
2
4
3
4
1
=B
5 4 3 2 0 0
S(B)
4 3 3 2 1 1
S'
2
4
3
4
1
5 4 3 1 0 1
S(B)
4 3 3 2 1 1
S'

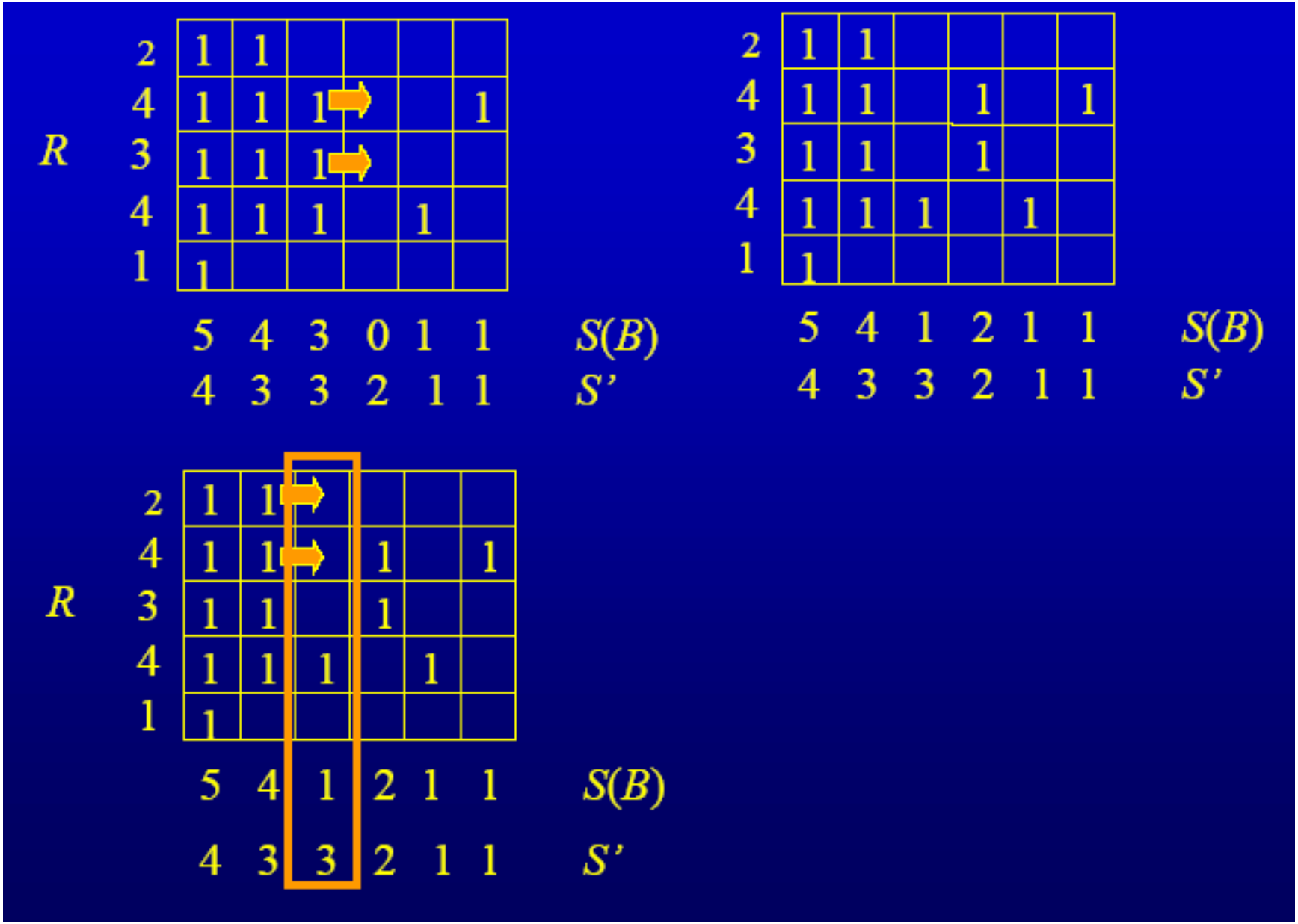

R
2
4
3
4
1
3 4 3 2 1 1
S
2
4
3
4
1
4 3 3 2 1 1
S'
R
2
4
3
4
1
1 1
1 1 1 1
1 1 1
1 1 1 1
1
=B
5 4 3 2 0 0
S(B)
4 3 3 2 1 1
S'
2
4
3
4
1
1 1
1 1 1 1
1 1 1
1 1 1 1
1
5 4 3 1 0 1
S(B)
4 3 3 2 1 1
S'

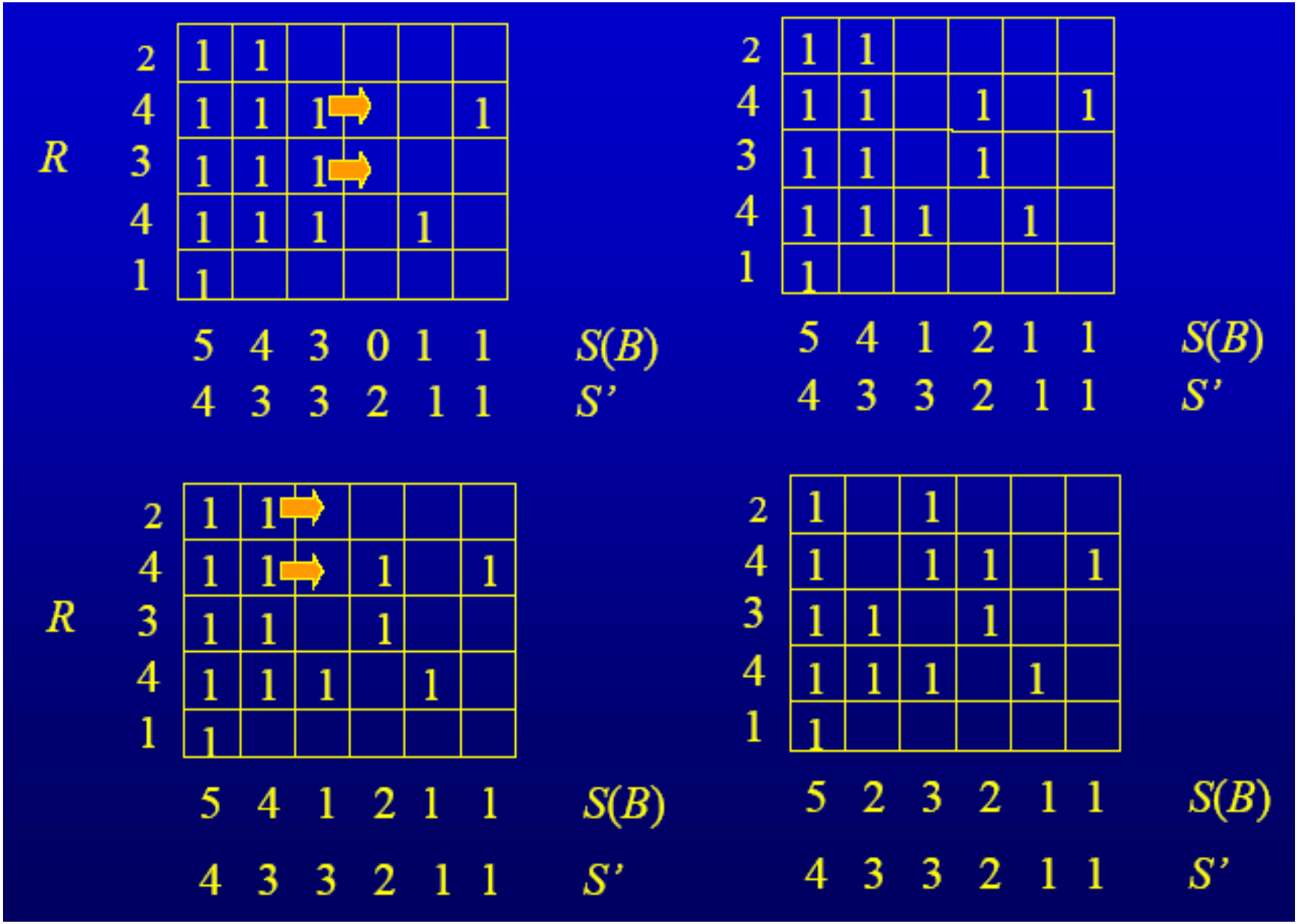

R
2
4
3
4
1
1 1
1 1 1 1
1 1 1
1 1 1 1
1
5 4 3 0 1 1 S(B)
4 3 3 2 1 1 S'

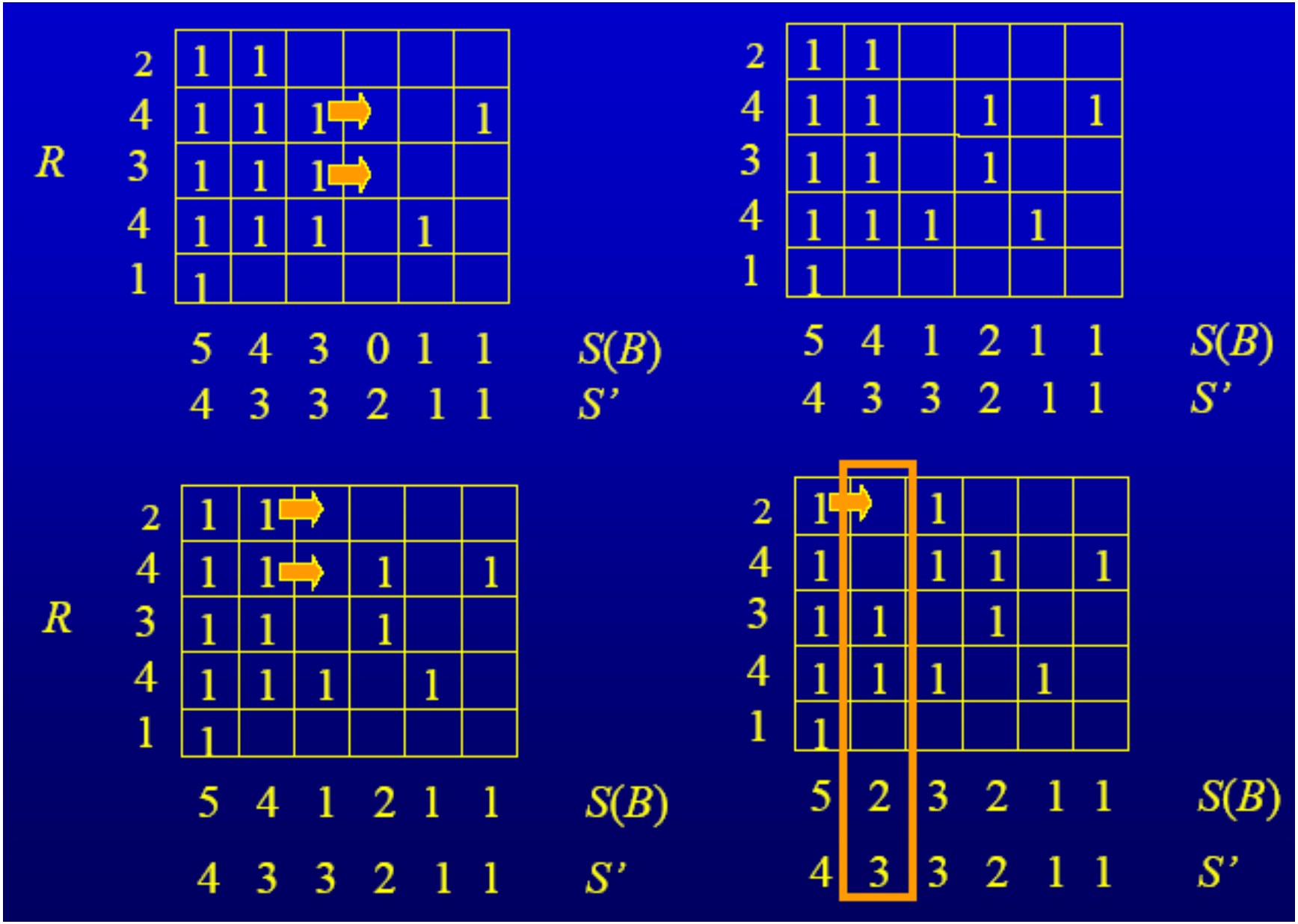

R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 3 0 1 1 S(B)
4 3 3 2 1 1 S'

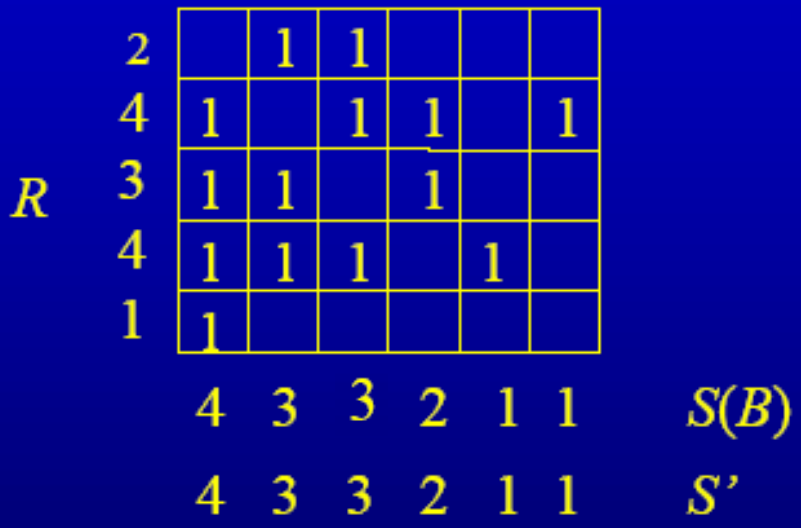

R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 3 0 1 1 S(B)
4 3 3 2 1 1 S'

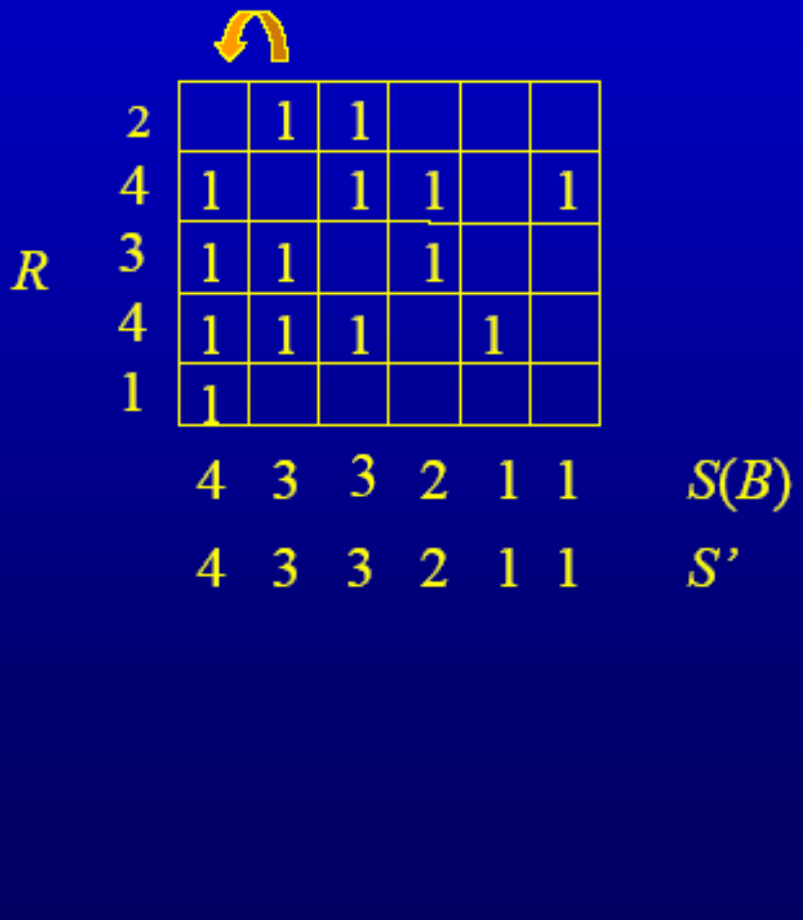

2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 1 2 1 1 S(B)
4 3 3 2 1 1 S'

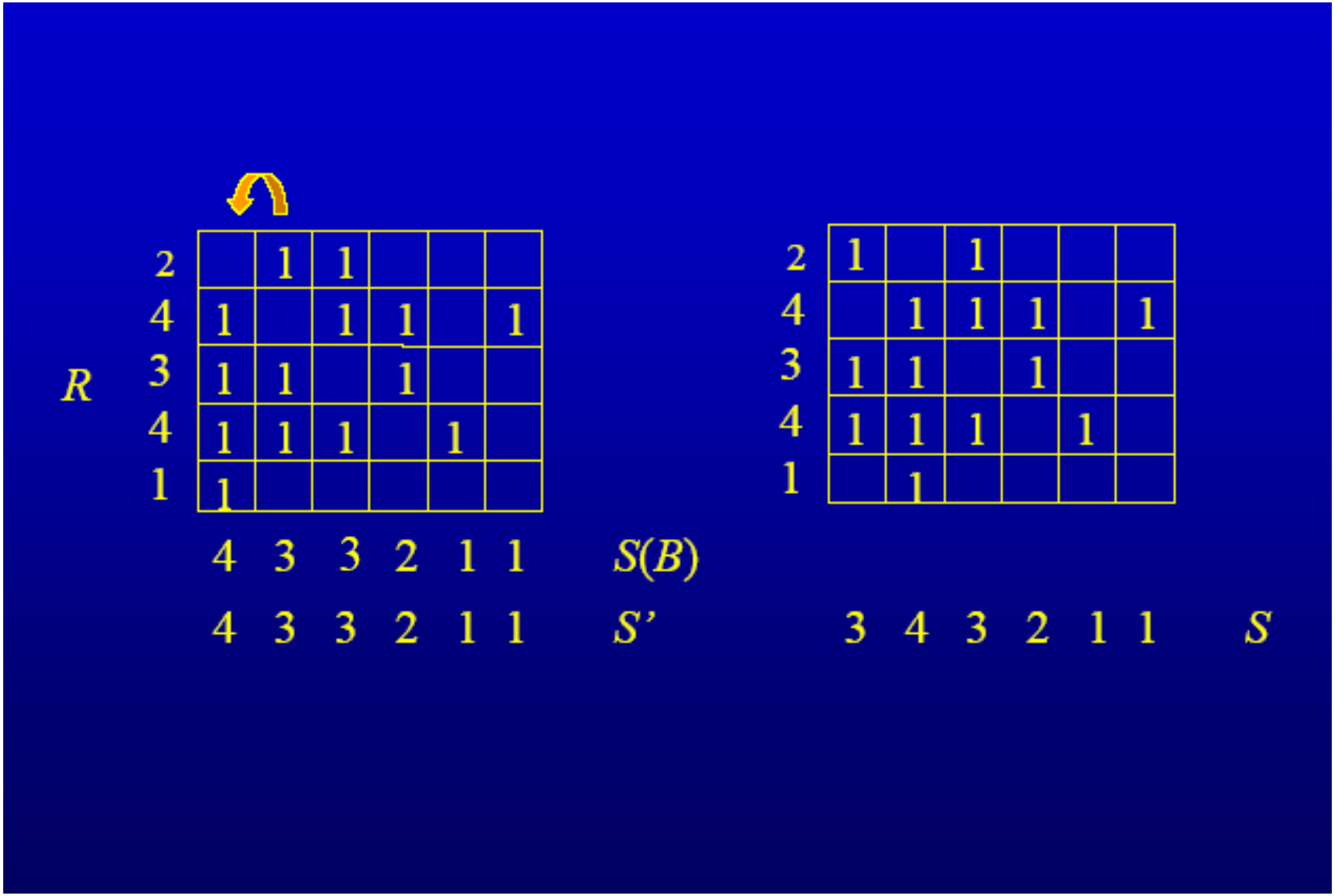

R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 3 0 1 1 S(B)
4 3 3 2 1 1 S'
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 1 2 1 1 S(B)
4 3 3 2 1 1 S'
R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 1 2 1 1 S(B)
4 3 3 2 1 1 S'


R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 3 0 1 1 S(B)
4 3 3 2 1 1 S'
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 1 2 1 1 S(B)
4 3 3 2 1 1 S'
R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 1 2 1 1 S(B)
4 3 3 2 1 1 S'
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 2 3 2 1 1 S(B)
4 3 3 2 1 1 S'

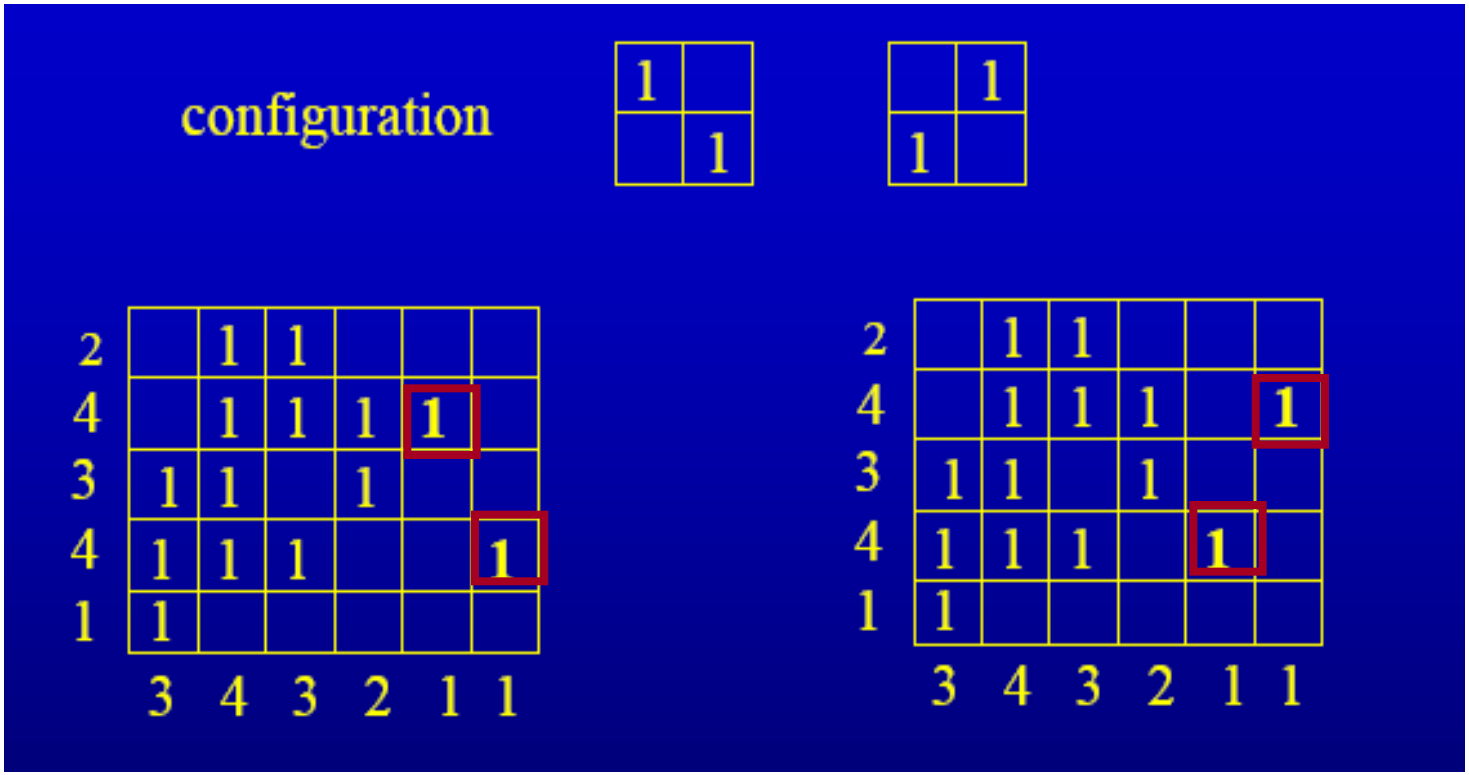

R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 3 0 1 1 S(B)
4 3 3 2 1 1 S'
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 1 2 1 1 S(B)
4 3 3 2 1 1 S'
R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 4 1 2 1 1 S(B)
4 3 3 2 1 1 S'
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
5 2 3 2 1 1 S(B)
4 3 3 2 1 1 S'


R
2
4
3
4
1
1 1
1 1 1
1 1 1
1 1 1 1
1
4 3 3 2 1 1
S(B)
4 3 3 2 1 1
S'


R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
4 3 3 2 1 1 S(B)
4 3 3 2 1 1 S'


R
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
4 3 3 2 1 1 S(B)
4 3 3 2 1 1 S'
2 1 1
4 1 1 1 1
3 1 1 1
4 1 1 1 1
1 1
3 4 3 2 1 1 S

# Kapcsoló komponensek és unicitás




A kapcsoló komponens jelenléte szükséges és elegendő a nem-egyértelműséghez