

## Tíz kérdés (amelyből 3–5-re kell majd válaszolnod laboron, papíron)

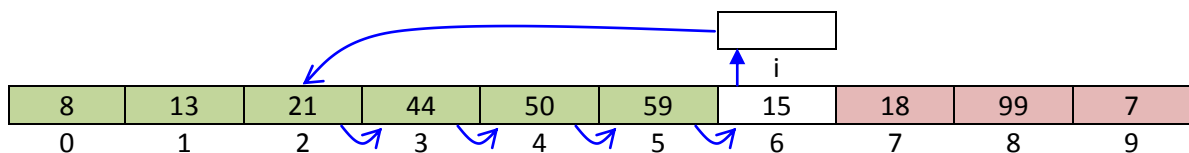
1. Mennyi a idő-bonyolultsága (komplexitása) a kiválasztásos rendezésnek?  $O(\dots)$
2. Mennyi a kiválasztásos rendezés komplexitása “legjobb esetben”?  $O(\dots)$
3. Mennyi a idő-bonyolultsága (komplexitása) a beszűrásos rendezésnek?  $O(\dots)$
4. Mennyi a beszűrásos rendezés komplexitása “legjobb esetben”?  $O(\dots)$
5. Mennyi a idő-bonyolultsága (komplexitása) a buborékos rendezésnek?  $O(\dots)$
6. Mennyi a buborékos rendezés komplexitása “legjobb esetben”?  $O(\dots)$
7. Mennyi a idő-bonyolultsága (komplexitása) a kiválasztásos rendezésnek?  $O(\dots)$
8. Hányszor hajtódik végre a 11. előadás 4. slide-ján lévő algoritmus `if` utasítása? Ez hány elemi műveletet (összehasonlítás/értékkadás) jelent, amennyiben a rendezendő számsorozat eredetileg növekvő/csökkenő?
9. Mennyi a idő-bonyolultsága (komplexitása) a lineáris keresésének?  $O(\dots)$
10. Mennyi a idő-bonyolultsága (komplexitása) a bináris keresésének?  $O(\dots)$

## LABOR–feladatok (Debugging!!!)

1. Implementáld a kiválasztásos rendezést úgy, hogy a belső ciklus maximumot keressen az  $a[i..n-1]$  szakaszon, majd e legnagyobb elemet cserélje ki  $a[i]$ -vel.
2. Implementáld iteratíván a beszűrásos rendezést az eltáncolt koreográfia alapján.
3. Implementáld iteratíván a buborékos rendezést úgy, mint aki arra számítász, hogy minden menetben csak a legnagyobb elem kerül a helyére.
4. Implementáld iteratíván a bináris keresését.

## HÁZI–feladatok:

1. Amit nem sikerült megoldani laboron.
2. Implementáld a beszűrásos rendezést úgy, hogy az  $a[i]$  elem beszűrásakor, az  $a[0..i-1]$  rendezett szakaszba, eltolásokat alkalmazol:



3. Implementáld iteratíván a buborékos rendezést az eltáncolt koreográfia alapján.
4. Implementáld iteratíván a shell rendezést.

## **11. feladatsor közép-haladóknak:**

1. Implementált a gyors-rendezést (quick-sort) (lásd google).
2. Implementált a merge-sort algoritmust (lásd google).