

2018. 10. 13.

2. Tanulmány

Algoritmusok 25.

Előadás-jellegű kurzus felvételéről.

algoritmus fogalma: Input, jól def. szám. eljárás, output.

pl.: legrendezett út keresése

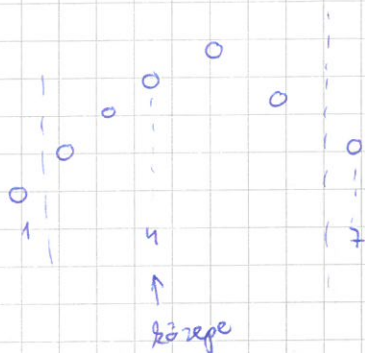
adatszerkezet fogalma: tárolás, szervezés, hozzáférés, módosítás...

csúcskereső algoritmus van.

1 db return van.

(Csúcskeresés: $O(N)$).

csúcskeresés



felismerés algoritmus is van na, ahogy néha már használt.

$O(\log n)$ volt, ugye.

2D csúcskeresés: $n \times n$ -es "matrixban".

mohó algoritmus: indultunk valahonnan és lépünk a nagyobb számok felé,

ha van. Ez $O(N^2)$.

1D-s keresésnek vezetjük vissza.

Van jó felismerés ebben a sorban

Maximum keresés (eredetileg benn): n .

⊕ felismerés: $O(n \log n)$. aszimptotikus felső korlát.

"Osid meg és analógia!" - algoritmus:

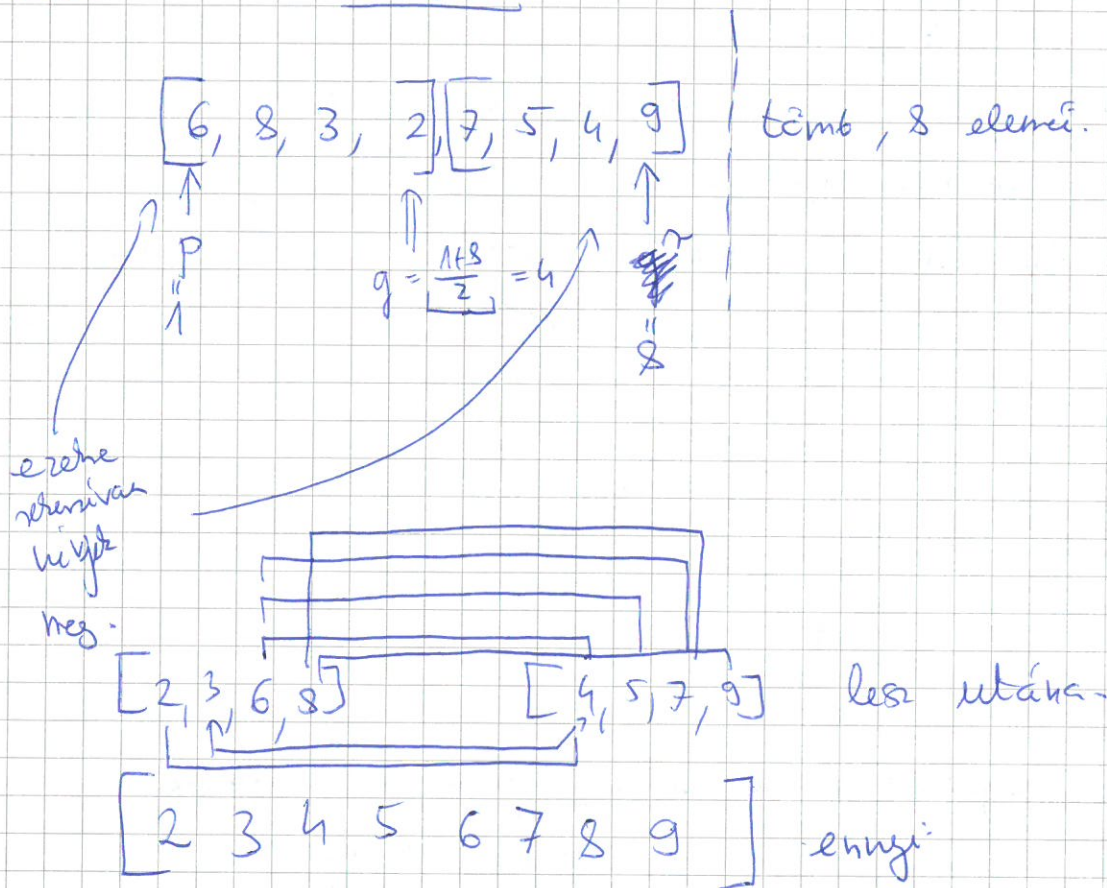
Rekurzív megoldás.

Rekurzív: jobban olvasható.

Intuitívabb.

felosztás van itt is, $O(\log n)$.

Kendézéssel: rekurzív összefésülő rendszer



Az összefésülés lineáris.

Felosztás konstans növekedési igényű.

elvégeztés: ...

[6, 8] [3, 2]

[6 8 3 2] [7 5 4 9]

[6 8 3 2]

[7 5 4 9]

[6 8] [3 2]

[7 5] [4 9]

6 8 3 2

7 5 4 9

②

es $O(\log n)$.

utána összekeverés:

[6, 8]

[2, 3]

[5, 7]

[4, 9]

[2, 3, 6, 8]

[4, 5, 7, 9]

[2, 3, 4, 5, 6, 7, 8, 9]

$2 \cdot \frac{db}{2} - 1$

n elemet
fésűként
minden
szára!

$O(n \cdot \lg n)$ les.
emiatt

Dinamikus programozás:

pénzfeladási feladat van.

Lehetne másképp algoritmus is.

De most nem.

Optimalizálási probléma. Van.

dinamikus esetben (ugye) lentől építkezünk felfelé.

dinamikus eset: $O(n)$ időben

iteratív megvalósítás. Ki's p. $\rightarrow$ nagy felé.

