Optimális bináris keresőfa

Adott különböző kulcsok egy $k_1 < k_2 < \dots < k_n$ kulcsok rendezett sorozata, ezekből szeretnénk bin. keresőfát építeni.
Ha minden $\boxed{k_i}$ kulcsnak ismert a keresési valószínűsége, akkor egy T bin. keresőfára megállapítható a keresés költségének várható értéke.
Egy kulcs megtalálásának a költsége legyen a keresés során megvizsgált kulcsok száma (a kulcs mélysége a fában)+1
A keresés költségének várható értéke: $\sum_{i=1}^{n} (d_T(k_i)+1) p_i$

<u>Feladat</u>: egy olyan bin. kerfa építése, amelynek a keresés költségének várható értéke minimális

DP. mo

— $k_1, k_2, \dots, k_n$ kulcsokat tartalmazó bináris ker. fa bármely részfája pontosan $k_i, k_{i+1}, \dots, k_j$ kulcsokat tartalmazza ($1 \le i \le j \le n$) (inorder bejárás)

- jelölje $C[i,j]$ a $\sum_{s=i}^{j} (d_T(k_s)+1)\, p_s$ értéket, amikor $k_i, k_{i+1}, \dots, k_j$ kulcsokból építünk bin. keresőfát
- egyszerűség kedvéért $C[i, i-1] = 0$ ($\forall i \in [1..n]$)

— Legyen T egy optimális megoldása ennek a feladatnak, amikor $k_i, k_{i+1}, \dots, k_j$ kulcsokból építjük a bin. kerfát

→ a gyökérben $\boxed{k_r}$ kulcs

- igazolható, hogy T baloldali T' részfája optimális m.o.: $k_i, \dots, k_{r-1}$-nek
  T'' jobboldali részfa $k_{r+1}, \dots, k_j$-nek (optimális részstr. tul.)

– rekurzív m.o:

$$C[i,j] = C[i,k-1] + \sum_{s=i}^{k-1} p_s + p_k + C[k+1,j] + \sum_{s=k+1}^{j} p_s =$$

$$= C[i,k-1] + C[k+1,j] + \sum_{s=i}^{j} p_s$$

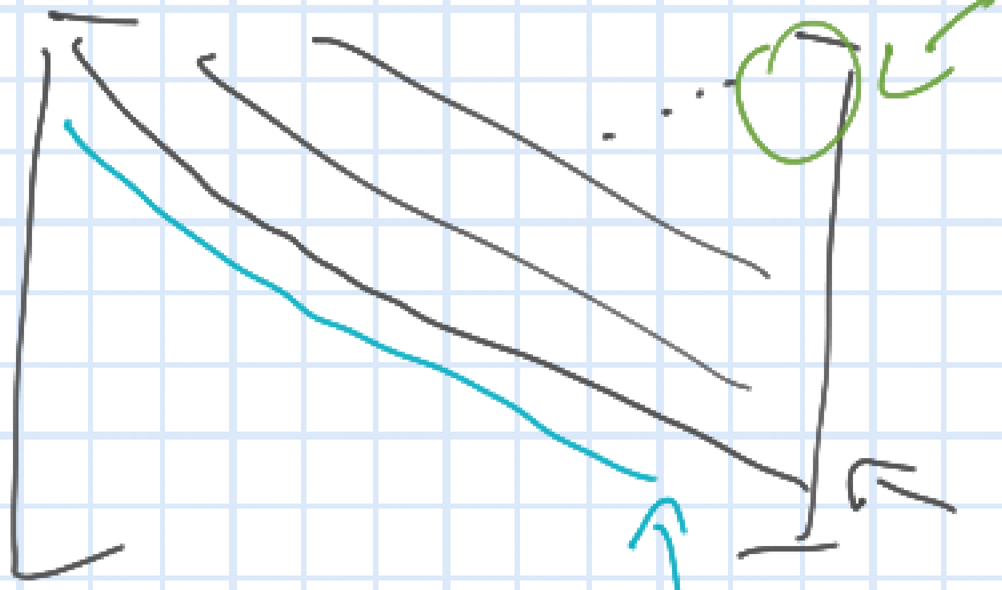
T' és T'' minden csúcsának mélysége eggyel nagyobb T-ben (e_k miatt)

- $k = i, i+1, \dots, j$ lehet
  ⇒ ezért végig kell vizsgálni

$$C[i,j] = \min_{i \le k \le j} \left\{ C[i,k-1] + C[k+1,j] + \sum_{s=i}^{j} p_s \right\}$$

– először $c[i, i-1] = 0$ értékadásokat végezzük el, majd a főátlót töltjük ki, és az ezzel párhuzamos átlókat

$c[i,i]$, $c[i,i+1]$, $c[i,i+2]$, ...



– egy optimális m.o.:

- legyen $d[i,j]$ a $k_i, \ldots, k_j$ kulcsokból álló opt. bin. keresőfa gyökerének a kulcsa

→ a fa rekurzívan felépíthető

– Költség: $O(n^3)$

Szekvenciák hasonlósága

— Pl.: két élőlény DNS láncának összehasonlítása

— két lánc lehet hasonló, ha:

- egyik a másiknak részstringje
- kevés változtatással átvihetők egymásba
- részsorozatok

Leghosszabb közös részsorozat

Egy $Z = (z_1, z_2, \dots, z_k)$ sorozat részsorozata egy $X = (x_1, x_2, \dots, x_n)$ sorozatnak, ha létezik olyan $1 \le i_1 < i_2 < \dots < i_k \le n$ indexek, hogy minden $1 \le j \le k$ esetén $z_j = x_{i_j}$.

Azt mondjuk, hogy egy Z sorozat közös részsorozata X és Y sorozatnak, ha Z részsorozata X-nek és Y-nak is.

Feladat: adjunk hatékony algoritmust az $X=(x_1,x_2,\dots,x_n)$ és $Y=(y_1,y_2,\dots,y_m)$ sorozatok egy leghosszabb közös részsorozatának meghatározására!

- **prefix**: az $X=(x_1,\dots x_n)$ sorozat $X_i=(x_1,x_2,\dots,x_i)$ részsorozatát az X i-edik prefixének nevezzük ($0\le i\le n$)

- optimális m.o. szerkezete
  - ha $x_n=y_m$, akkor $z_k=x_n=y_m$ és Z_{k-1} leghosszabb közös részsorozata X_{n-1}-nek és Y_{m-1}-nek
  - ha $x_n\neq y_m$ és $z_k\neq x_n$, akkor Z egy leghosszabb közös részsorozata X_{n-1}-nek és Y-nak
  - ha $x_n\neq y_m$ és $z_k\neq y_m$, akkor Z lhkr X-nek és Y_{m-1}-nek

– Egy vagy 2 részfeladat van, amit meg kell vizsgálni

- ha $x_n = y_m$, akkor X_{n-1} és Y_{m-1} egy l.h.k.r.-t kell megtalálni, amelyhez hozzávéve $x_n = y_m$ elemet X és Y l.h.k.r.-t kapjuk
- ha $x_n \neq y_m \Rightarrow$ 2 részfeladat

  $\left.\begin{array}{l} \to X_{n-1} \text{ és } Y \\ \to X \text{ és } Y_{m-1} \end{array}\right\}$ ezek közül a hosszabbat vesszük

– rekurzív m.o.:

- X_i és Y_j l.h.k.r.-nak hosszát jelölje $c[i,j]$ $0 \leq i \leq n$, $0 \leq j \leq m$

  ↳ ha $i=0 \vee j=0 \Rightarrow c[i,j]=0$

$$c[i,j] = \begin{cases} 0, & \text{ha } i=0 \vee j=0 \\ c[i-1,j-1]+1, & \text{ha } i,j>0 \text{ és } x_i = y_j \\ \max(c[i-1,j], c[i,j-1]), & \text{ha } i,j>0 \text{ és } x_i \neq y_j \end{cases}$$

- egy opt. m.o.: $1 \leq i \leq n$ és $1 \leq j \leq m$ esetén $b[i,j]$ a c tömb azon elemének indexe, amelyen $c[i,j]$ meghatározásakor a legjobb értéket kaptuk
- c-t sorfolytonosan töltjük ki, soronként balról jobbra
- Költség: egy cella $O(1)$
  összesen: $O(nm)$

megoldás $c[n,m]$-ben

- $b[n,m]$-től indulva, a „mutatók" mentén haladunk
  - ha $b[i,j]$ elemről $b[i-1,j-1]$ elemre lépünk, akkor $x_i = y_j$ benne van a l.h.k.r.-ban

  → fordított sorrendben kapjuk meg az elemeket

$X = (a, b, a, b, b, a, b, b, a)$

LHKR: a b a b a b

$Y = (b, a, a, b, a, b, a, b)$

		b	a	a	b	a	b	a	b
	0	0	0	0	0	0	0	0	0
a	0	↑ 0	↖ 1	↖ 1	← 1	↖ 1	← 1	↖ 1	← 1
b	0	↖ 1	↑ 1	↑ 1	↖ 2	← 2	↖ 2	← 2	↖ 2
a	0	↑ 1	↖ 2	↖ 2	↑ 2	↖ 3	← 3	↖ 3	← 3
b	0	↖ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4	← 4	↖ 4
b	0	↖ 1	↑ 2	↑ 2	↖ 3	↑ 3	↖ 4	↑ 4	↖ 5
a	0	↑ 1	↖ 2	↖ 3	↑ 3	↖ 4	↑ 4	↖ 5	↑ 5
b	0	↖ 1	↑ 2	↑ 3	↖ 4	↑ 4	↖ 5	↑ 5	↖ 6
b	0	↖ 1	↑ 2	↑ 3	↖ 4	↑ 4	↖ 5	↑ 5	↖ 6
a	0	↑ 1	↖ 2	↖ 3	↑ 4	↖ 5	↑ 5	↖ 6	↑ 6