Algoritmuselmélet
4. heti feladatsor megoldások

1. Módosítsa az órán tanult gyorskeresés mintaillesztő algoritmust úgy, hogy az ugrófüggvényt két karakterre definiáljuk, azaz $U(x)$ helyett $U(xy)$ értékeket határozunk meg minden lehetséges xy karakterpárra az ábécéből a minta előfeldolgozása során és ezt az ugrófüggvényt használjuk később az illesztés során. (Itt U jelentése ugyanaz, mint az eredeti változatban, csak annyi a különbség, hogy nem egy, hanem két karaktert tekintünk.)
(a) Írja le az ugrófüggvény meghatározásának algoritmusát.
(b) Írja le, hogy hogyan használjuk ezt az ugrófüggvényt a minta keresése során és röviden (1-2 mondatban) indokolja meg, hogy ez a módszer miért helyes.
(c) Tekintsük a 100 darab A betűből álló szöveget. Adjon meg egy olyan, legalább 4 hosszú mintát az $\{A, B, C, D\}$ ábécé felett, amire az új ugrófüggvénnyel készített algoritmus pontosan ugyanannyi összehasonlítást használ, mint az órán tanult eredeti változat, amikor a minta összes előfordulását keressük a szövegben.
(d) Tekintsük a 100 darab A betűből álló szöveget. Adjon meg egy olyan, legalább 4 hosszú mintát az $\{A, B, C, D\}$ ábécé felett, amire az órán tanult eredeti változat legalább ötször olyan sok összehasonlítást használ, mint ez az új változat, amikor a minta összes előfordulását keressük a szövegben.

Megoldás
(a) Az ugrófüggvény $U[xy]$ értéke azt adja meg, hogy az xy részszó utolsó előfordulása hátulról nézve hányadik karakternél végződik az m hosszú M mintában és $U[xy] = m$, ha xy nem szerepel a mintában sehol.
Az algoritmus az ugrófüggvény meghatározására:

- Először $U[xy] = m$ minden xy szóra.
- Ezután sorban $i = 1, 2, 3, \ldots, m-1$ értékekre $U[M[i]M[i+1]] = m - i$

(b) Az ugrófüggvényt így használjuk: először $k = 0$ eltolással kezdjük az illesztést és amikor ezt megvizsgáltuk (akár sikeres, akár sikertelen illesztés volt) $k+1$ helyett a $k + U[S[k+m]S[k+m+1]]$-es eltolás következik (vagy ha $S[k+m+1]$ már nem létezik, mert elfogyott a szó, akkor vége az algoritmusnak).

Az eljárás azért helyes, mert $U[S[k+m]S[k+m+1]]$-nél kisebb ugrás esetén az $S[k+m]S[k+m+1]$ szó (a szöveg a mostani illesztésben részt vevő utolsó és az ezután következő betűből álló kettő hosszú szó) a mintában hátrébb levő helyre kerülne, mint ahol valójában utoljára előfordul.

(c) Ha $M = AAAAA$, akkor mindkét eljárás minden eltolást megnéz $k = 0$-tól $k = 96$-ig, mindegyik esetben 5 összehasonlítást téve. Ez azért igaz, mert $U[A] = 1$ az eredeti változatban és $U[AA] = 1$ az új változatban, azaz a k-as eltolás után mindkét esetben a $k+1$-es jön majd.

(d) Tekintsük például a 20 hosszú $ABAB \ldots ABAB$ mintát (azaz ahol felváltva vannak A és B betűk). Ekkor az eredeti vátozatban $U[A] = 2$ miatt $k = 0, 2, 4, \ldots, 80$-as eltolásokat nézünk, mindegyik 2 összehasonlítás, azaz összesen $41 \cdot 2 = 82$ összehasonlítás. Az új változatban $U[AA] = 20$, azaz a $k = 0, 20, 40, 60, 80$-as eltolásokat nézzük, mindegyik 2 összehasonlítás, azaz ez összesen csak 10 összehasonlítás.

2. Tekintsük az alábbi döntési problémát:
Szomszédossági mátrixával adott egy G irányítatlan gráf. Azt kell eldönteni róla, hogy el lehet-e hagyni a gráfból legfeljebb 42 csúcsot úgy, hogy a maradék csúcsok kiszínezhetők legfeljebb 4 színnel úgy, hogy minden csúcsnak legfeljebb 7 vele azonos színű szomszédja van.

Indokolja meg részletesen, hogy ez a probléma miért van NP-ben.

Megoldás
A tanú tétel alapján azt kell megmutatni, hogy van olyan L_1 nyelv és vannak olyan c_1, c_2 konstansok, amikre igaz az, hogy

- egy x input pontosan akkor jó, ha van hozzá olyan y tanú, amire igaz, hogy $|y| \leq c_1|x|^{c_2}$ és $(x, y) \in L_1$
- $L_1 \in P$

Álljon az L_1 nyelv azon $(G, color)$ párokból, ahol G egy irányítatlan gráf, $color$ pedig egy olyan, a csúcsokkal indexelt tömb, ami a gráf csúcsainak egy olyan 4 színnel való színezését adja meg, amelyre:

- a csúcsok színe p, k, z, s vagy null

- legfeljebb 42 olyan csúcs van, amire a szín null
- minden olyan csúcsra, aminek van színe (azaz $color[v]$ nem null) igaz, hogy a szomszédai között legfeljebb 7 ugyanolyan színű csúcs szerepel

Ha egy input egy jó gráf, akkor a hozzá tartozó y tanú az a színezés, amiben az elhagyandó csúcsok színe null, a többi csúcs színe pedig a jó 4-színezés szerinti szín. Világos, hogy a jó gráfokra van ilyen tanú, a rossz gráfokra pedig nincsen.

A tanú hossza $O(n)$, mert *color* egy n méretű tömb, ahol minden bejegyzés konstans hosszú (5 lehetőség egyike). Mivel a input mérete n^2 (n csúcsú gráf szomszédossági mátrixának ez a mérete), ezért $|y| \leq c|x|$ vagyis a tanú rövid.

A $(G, color)$ pár L_1-be tartozásának ellenőrzésére azt kell eldönteni adott színezésről, hogy

- a null-on kívül csak 4 érték szerepel benne
- legfeljebb 42 null cella van
- minden v csúcsra v szomszédai között legfelebb 7 darab $color[v]$ színű csúcs van

Ezt ellenőrizni tudjuk így:

- egyszer végigmegyünk a tömbön és megszámoljuk a használt színeket, ez $O(n)$
- egyszer végigmegyünk a tömbön és megszámoltuk a null cellákat, ez is $O(n)$
- minden v csúcsra végigmegyünk G szomszédossági mátrixában v során és megszámoljuk, hogy hány $color[v]$ színű csúcs van, ez egy csúcsra $O(n)$, az összes csúcsra $O(n^2)$

A teljes ellenőrzés $O(n^2)$, azaz $O(|x|)$, vagyis ez az eljárás P-ben van.