ALGORITMUSOK ÉS BONYOLULTSÁGELMÉLET

Matematika MSc hallgatók számára

11. Előadás

Előadó: Hajnal Péter
Jegyzetelő: Hajnal Péter

2010. április 26.

$\mathcal{ZPP}$

A következő lemma egy nagyon fontos nyelvosztályt ír le különböző módokon.

1. Lemma. *Legyen L egy nyelv. A következők ekvivalensek:*

(i) $L \in \mathcal{RP} \cap co\mathcal{RP}$,

(ii) Van olyan véletlen polinomiális idejű T algoritmus (leálló állapotai ELFOGAD, ELVET és NEM-IS-TUDOM), amelyre teljesül

- $\omega \in L$ *esetén* $\mathbb{P}_\rho[T(\omega,\rho) = ELFOGAD] \geq 1/2$, *és* $\mathbb{P}_\rho[T(\omega,\rho) = ELVET] = 0$.
- $\omega \notin L$ *esetén* $\mathbb{P}_\rho[T(\omega,\rho) = ELVET] \geq 1/2$, *és* $\mathbb{P}_\rho[T(\omega,\rho) = ELFOGAD] = 0$.

(iii) Van olyan véletlen (nem szükségszerűen leálló) T algoritmus (leálló állapotai ELFOGAD, ELVET), amelyre teljesül

- *Van olyan $p(x)$ polinom, hogy minden $\omega \in \Sigma^*$ esetén*

$$\mathbb{E}[TIME(T(\omega,\rho))] \leq p(|\omega|),$$

- *$\omega \in L$ esetén T leállása esetén biztos ELFOGAD, míg $\omega \notin L$ esetén T leállása esetén biztos ELVET.*

Bizonyítás. (i)⇒(ii) A (ii) pont követelményeinek elegettevő algoritmust úgy kapjuk, hogy először az L-et eldöntő $\mathcal{RP}$ algoritmust futtatjuk. Ha ez elfogad, akkor leállunk ELFOGAD állapottal. Ha nem, akkor az L-et eldöntő co$\mathcal{RP}$ algoritmust futtatjuk. Ha ez elvet, akkor leállunk ELVET állapottal. Különben NEM-IS-TUDOM állapottal fejeződik be a számítás. Az előírt követelmények könnyen láthatóan teljesülnek.

(ii)⇒(iii) Futassuk az L-et eldöntő feltételben leírt algoritmust. Ha NEM-IS-TUDOM állapottal áll le, akkor ismételjük meg (persze a szükséges véletlen biteket a szalagról olvassuk — nem a korábbiakat használjuk — azaz az új futás független lesz az előzőtől). Az ismételést addig végezzük, amíg az egyik futásnál ELFOGAD vagy ELVET állpothoz jutunk. Az előírt követelmények közül csak a futási idó várhatóértékének becslése nem triviális. Elég az ismételések számának várható értékét becsülni.

Ezt $\omega \in L$ esetén végezzük el. Ha egy ismételést egy pénzfeldobásnak gondolunk, amelyben a FEJ az elfogadás, míg az ÍRÁS a nem-is-tudom állapot, akkor addig dobálunk, amíg FEJ nem jön ki (amely valószínűsége minden dobásnál ugyanaz az érték, legalább 1/2). A dobások függetlenek, a dobások számának várható értékének becslése standard valószámítási feladat, konstansnak adódik.

(iii)⇒(i) Legyen p a feltétel által garantált algoritmus futási idejére adott korlát polinomja. Szimuláljuk ezt az algoritmust $2p(|\omega|)$ lépésig. Ha közben leáll (elfogad vagy elvet), akkor mi is bejelentjük a szimuláció által kiszámolt végeredményt. Különben NEM-IS-TUDOM állapottal fejezzük be a számítást. A Markov-egyenlőtlenség adja, hogy ez utóbbi esemény valószínűsége legfeljebb 1/2. ■

Definíció. A fenti tulajdonságoknak (illetve bármelyiküknek) elegetttevő nyelvek osztályát $\mathcal{ZPP}$-vel jelöljük.

Megjegyzés. 1) Az osztályt adó betűszó Z betűje a hibázás lehetetlenségére utal (zero error), a két P betű a probabilistic és polynomial szavakból ered.

2) Az eddig megismert véletlen algoritmusok két osztályba esnek: garancia van futási idejükre, de végeredményük hibázást rejthet ($\mathcal{BPP}, \mathcal{RP}, \text{co}\mathcal{RP}$), illetve garancia van a végeredményükre, de futási idejük egy valószínűségi változó ($\mathcal{ZPP}$). A véletlen algoritmusokban rejlő fent említett két lehetőség megkülönböztetésére rendre a Monte Carlo, illetve Las Vegas jelzőket használják.

Kielégítő értékelés keresése kis fokú k-CNF formulához

Legyen φ egy k-CNF. Legyen Δ egy becslés, hogy egy klóz hány másikkal (saját magát beszámolva) tartalmaz közös változót (elképzelhető, hogy egyikben negálva másikban negálatlanul). Feltesszük, hogy $\Delta \leq 2^{k-2}$. Célunk, hogy konstruktívan belássuk, hogy φ kielégíthető. Egy $\mathcal{ZPP}$ algoritmust adunk meg, ami a feltételeknek elegettevő φ-hez kielégítő értékelést ad. Az algoritmus nagyon egyszerű lesz (analízise már nem annyira).

Kiindulás: `Változóinknak adjunk független, uniform eloszlású véletlen bitértékeket.`
Teszt: `Teszteljük, hogy kielégíti-e` φ`-t.`
Szerencse: `Ha igen, akkor leállunk az aktuális éltékadással.`
Szerencsétlenség, okkeresés: `Ha nem, akkor keressünk egy` C `klózt, ami nincs kielégítve.`
Szerencsétlenség, ÚJRASOROSOLÁS: `Értékadásunkat a` C`-ben szereplő változókon változtassuk meg. A` k `darab változónak az előzőektől független véletlen biteket adunk értékként.`
`VISSZA a Teszt lépéshez.`

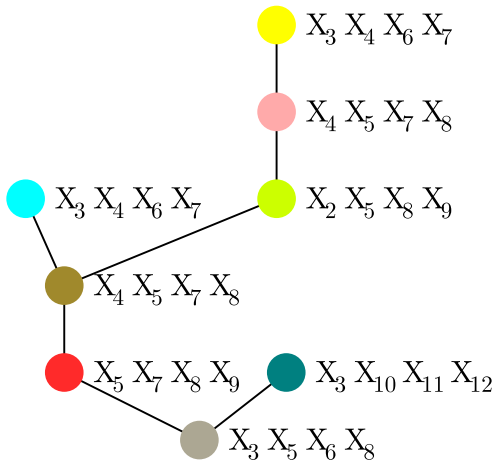
Ha belátjuk, hogy algoritmusunk bármilyen feltételünknek elegettevő k-CNF-re pozitív valószínűséggel leáll, akkor a kielégíthetőségre vonatkozó matematikai tételt beláttuk. Ha azt is igazoljuk, hogy a futás során az újrasorsolások számának várható értéke az input hosszának polinomjával becsülhető, akkor egy $\mathcal{ZPP}$-konstruktív bizonyítást adtunk.

A felhasznált véletlen biteket egy táblázatban helyezzük el. A táblázat oszlopai a változókkal vannak azonosítva. Végtelen sok sora van, amelyek természetes számokkal indexeltek. Ezt a táblázatot töltjük ki független bitekkel. Ha algoritmusunknak véletlen bitekre van szüksége, akkor ezeket a táblázatból olvassa ki. Egy változó értékadásánál az oszlopából az első nem olvasott bitet adjuk értékként.

A 0 indexű sor tartalmazza a kiinduló értékadás bitjeit. Minden újrasorsolás k bit olvasását jelenti: az újrasorsolt változók oszlopában az utolsónak kiolvasott bit alatti bit elővétele történik. Jelölje b_ℓ, B_ℓ és V_ℓ az ℓ-edik újrasorsolás k bitjét, ezek pozícióit a táblázatban (ezeket blokkoknak nevezzük), illetve azon változók halmazát, amelyeket újrasorsoltunk. A különböző blokkok természetesen diszjunkt pozícióhalmazok. Az első újrasorsolás értékei az 1 indexű sorból kerülnek ki. A későbbi blokkok azonban a sorok tekintetében töredezettek lehetnek, akár az is elképzelhető, hogy ugyanazon újrasorsolás k bitje k különböző sorból vevődik. Ha B_i és B_j két blokk és az elsőnek van olyan bitje, ami közvetlen a második (B_j) egy bitje felett van, akkor azt mondjuk, hogy B_i fedi B_j-t. Természetesen ekkor $i < j$, a két blokk diszjunkt és ha ugyanazon oszlopból mindkettőnek van bitje, akkor az B_i-beli van magasabb pozícióban.

Annak a ténynek, hogy az ℓ-edik újrasorsolás megtörtént megfeleltethetünk egy O_ℓ oksági fát. A fa gyökere V_ℓ lesz. A felépítését az i-edik újrasorsolások vizsgálatával, $i = \ell - 1, \ell - 2, \ldots, 2, 1$ visszamenő sorrendben, rekurzióval építjük fel. Ha az i-edik újrasorsolás blokkja fedi a fa már egy meglévő csúcsának blokkját, akkor ezek közül a legmélyebb fiaként a fához csatolunk egy csúcsot, amit V_i-vel címkézünk. Ha ilyen blokk nem volt, akkor fánkat nem bővítjük, továbblépünk. Az utolsó vizsgálat (ami az első újrasorsolásra vonatkozik) után alakul ki a teljes oksági fa.

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}
0	1	1	0	1	1	1	0	0	1	1	0
1	0	1	0	1	1	0	1	0	1	0	0
1	1	0	0	1	1	1	0	0	1	1	0
0	1	0	0	1	1	0	1	1	0	0	1
0	1	1	0	0	1	1	1	0	0	0	0
1	0	1	0	1	0	0	1	0	0	1	1
0	0	0	0	0	1	0	0	1	1	0	1
1	1	0	1	0	1	0	0	0	0	1	1
0	1	0	0	1	1	0	1	1	0	1	1
1	1	1	0	0	1	0	0	1	1	0	0
0	0	1	1	0	1	1	0	0	0	0	1



1. ábra. Algoritmusunk egy futása (azonos szín azonos blokk) és a szürke blokk oksági fája

Két nagyon fontos megjegyzést teszünk:

- Vegyük észre, hogy ha két blokk ugyanazon klóz miatt lett újrasorsolva, akkor

a megfelelő pozíciók felett (a klóz változóinak régi értékadása) ugyanazok a bitek szerepelnek, azok amik a kérdéses klózt hamissá teszik (CNF formulánál ez a klózban szepelő változók egyetlen értékadására teljesül).

- A fából (a csúcsok mellett a változó k-asok címkéivel) visszafejthető, hogy a fa csúcsainak megfelelő egyes újrasorsolások során a változók hanyadszorra lettek újrasorsolva, azaz a fa csúcsaihoz tartozó blokkok táblázatbeli helyzete.

Legyen $E_{\geq dn}$ az az esemény, hogy több mint $d \cdot n$ újrasorsolás történt. Ekkor a blokkokban lesz olyan bit, amit legalább d indexű sorba esik. Ekkor ennek oksági fája legalább d csúcsot tartalmaz (sőt legalább d lesz a mélysége).

Legyen T ezen blokk oksági fája. Legyen E_T azon esemény, hogy az algoritmus futása során valamelyik újrasorsolás redukált oksági fája T lesz.

Így az $E_{\geq dn}$ eseményt lefedhetjük az $\cup_{T:|V|\geq d} F_T$ eseménnyel, ahol F_T az az esemény, hogy a futás során valamelyik újrasorsolás/blokk oksági fája T.

Legyen T egy v darab csúcsot tartalmazó oksági fa. Ekkor

$$\mathbb{P}(F_T) \leq \left(\frac{1}{2^k}\right)^v = \frac{1}{2^{kv}},$$

hiszen minden csúcs azt jelenti, hogy a megfelelő klózt a korábbi véletlen bitek nem elégítették ki (amely bit k-asok a táblázat diszjunkt pozícióhalmazaiból jöttek). Tehát v darab független esemény egyszerre történő bekövetkezése, amelyeknek egyenként a valószínúsége $1/2^k$.

Másrészt felülről becsülhetjük hány t pontú oksági fa van. Legyen m a φ formula klózainak száma. Az oksági fa gyökerét m-féleképpen választhatjuk. Ezekután $v-1$ águhajtást hajtunk végre. Mindegyiknél meg kell mondanunk, hogy melyik csúcsból (v darab lehetőség) melyik másikhoz vezet az águhajtás. Ez legfeljebb Δ lehetőség, hiszen a szomszéd klózjában szerpló változókat kell csak leírnunk, amelyek halmaza metszi a csúcs változóhalmazát. Ilyen klóz csak Δ lehet. Így az águhajtások lehetőségét $v\Delta$-val becsülhetjük felül, amiből $v-1$ valósul meg. A v pontú oksági fák száma legfeljebb $m\binom{v\Delta}{v-1}$ Azonban

$$m\binom{v\Delta}{v-1} \leq m\binom{v\Delta}{v} \leq m(e\Delta)^v.$$

Így

$$\mathbb{P}(E_{\geq dn}) \leq \mathbb{P}(\cup_{T:|V|\geq d} F_T) \leq \sum_{v=d}^{\infty} (e\Delta)^v \cdot \frac{1}{2^{kv}} = m\frac{\left(e\Delta/2^k\right)^d}{1 - e\Delta/2^k}.$$

Esetünkben

$$\frac{e\Delta}{2^k} \leq \frac{e2^{k-2}}{2^k} \leq \frac{3}{4}.$$

Azaz

$$\mathbb{P}(E_{\geq dn}) \leq 4m\left(\frac{3}{4}\right)^d.$$

Ezekután jelöljük ξ-vel az újrasorsolások számát megadó valószínűségi változót. Várható értéke most már könnyen becsülhető:

$$\begin{aligned}\mathbb{E}(\xi) &= \textstyle\sum_{i=0}^{\infty} i \cdot \mathbb{P}(\xi = i) = \sum_{i=1}^{\infty} \mathbb{P}(\xi \geq i) = \sum_{j=1}^{\infty} n\mathbb{P}(\xi_j \geq (j-1)n) = \\ &= \textstyle\sum_{j=1}^{\infty} n\mathbb{P}(E_{\geq (j-1)n}) \leq \sum_{j=0}^{\infty} n \cdot 4m\left(\frac{3}{4}\right)^j = 16nm.\end{aligned}$$

Ami nyilván polinomiális φ méretében.

Összefoglalva:

2. Tétel. *Legyen φ egy $k-CNF$ formula, amelyben minden klóz legfeljebb 2^{k-2} másikkal tartalmaz közös változót (belértve magát is). Ekkor φ kielégíthető, sőt van olyan a futási idő várható értékében polinomiális algoritmus, ami megtalál egy kielégítő értékadást.*

Megjegyzés. Természetesen a 2^{k-2}-es becslés helyettesíthető $2^k/(e+\epsilon)$-nal, tetszőleges $\epsilon > 0$ konstansra.

Megjegyzés. A tétel egzisztenciális részének szokásos bizonyítása Lovász László lokális lemmájának standard alkalmazása. Az algoritmikus megoldás sokáig nyitott volt és a jelen algoritmus egy hosszú kutatásra tesz koronát. A tétel Robin Moser és Tardos Gábor 2009-ben bejelentett eredményének egy kissé egyszerűsített változata.

Irányítatlan gráfokra vonatkozó elérhetőség

Definíció. Az IRÁNYÍTATLAN-ELÉRHETŐSÉG problémában adott egy G irányítatlan gráf és két kitüntetett csúcsa s és t. El kell döntenünk, hogy van-e G-ben út a két kitüntetett csúcs között.

A probléma megoldásához egy véletlen sétát végzünk s-ből indulva. Ha közben t-t elérjük, akkor tudjuk, hogy s és t között van út. Ha elég hosszú séta során sem érjük el t-t, akkor elég nagy bizonyossággal sejtjük, hogy s és t között nincs út.

Kiindulás: `Legyen aktuális-csúcs az` s `csúcs.` $\ell = 0$`, az eddig megtette lépések száma.`
Véletlen lépés: `Az aktuális-csúcs szomszédai közül válasszunk ki egy véletlen` r `csúcsot. Legyen aktuális-csúcs` r `és legyen` $\ell = \ell + 1$.
Teszt: `Ha aktuális-csúcs=` t`, akkor ELFOGAD állapottal leállunk.`
`Ha` $\ell < T$`, akkor a véletlen lépést hajtjuk végre.`
`Ha` $\ell \geq T$`, akkor a VALÓSZÍNŰLEG-ROSSZ állapottal leállunk.`

Ebben a pillanatban az algoritmus leírása egy kissé pontatlan: nem írtuk le pontosan a véletlen lépést, illetve szerepel egy egyelőre definialatlan T paraméter. Ezekre a későbbiekben térünk ki.

Az egyetlen mód, ahogy algoritmusunk hibázhat, az az ha s és t a G gráf egye komponensébe esik és az T hosszú véletlen séta elkerüli t-t. Belátjuk, ha $T = 2|V|^3$, akkor ennek valószínűsége kisebb mint $1/2$. Ezzel kapjuk, hogy IRÁNYÍTATLAN-ELÉRHETŐSÉG$\in \mathcal{RP}$. Igazából algoritmusunk tárigénye logaritmikus. A megfelelő osztály bevezetésénél óvatosnak kell lennünk.

Definíció. Egy L nyelv akkor és csak akkor tartozik az $\mathcal{RL}$ osztályhoz, ha polinomiális idejű, logaritmikus tárat használ és az (R_1) és (R_2) feltételeket teljesíti.

Így a következő eredményt kapjuk.

3. Tétel. *IRÁNYÍTATLAN-ELÉRHETŐSÉG$\in \mathcal{RL}$.*

A bizonyítás következő erdményből nyilvánvaló.

4. Tétel. *Legyen G egy n pontú irányítatlan gráf. Ekkor tetszőleges pontjából indítva egy véletlen sétát annak a lépésszámnak, amely az összes csúcs meglátogatásához szükséges a várható értéke kisebb mint $|V|^3$.*

Bizonyítás. Nézzük meg. hogy véletlen sétánk L lépése után milyen eloszlású lesz helyzetünk (a V csúcshalmaz felett). Célunk, hogy belássuk az eloszlás konvergál egyetlen eloszláshoz (határeloszláshoz). Ez általában nem igaz. Ha G páros gráf, akkor L paritása alapján az eloszlások különböző színosztályokra koncentrálódnak. A határeloszlás is egyszerűbb lesz, ha ügyesen írjuk le véletlen lépésünket.

Legyen A a gráfunk szomszédsági mátrixa. Legyen Δ a gráf maximális foka. A főátlóba írjunk olyan számokat, hogy minden sor és oszlop összeg $\Delta + 1$ legyen. Jelöljük $\widetilde{A}$-mal azt a mátrixot, amit így kaptunk. Legyen M az a mátrix amit $\widetilde{A}$ elemeinek $\Delta + 1$-gyel való osztásával kapunk. Az így kapott mátrix szimmetrikus, nem negatív, főátlójában pozitív értékű és minden sor-, oszlopösszege 1 (úgy nevezett *duplán sztochasztikus mátrix*).

Az M mátrix egy véletlen séta átmeneti mátrixaként értelmezhető. Ez lesz a véletlen sétánk. Ez persze elemi módon is leírható. Egy v csúcsnak $d(v)$ szomszédja van, amiket azonosíthatunk az $\{1, 2, \ldots, d(v)\}$ halmaz elemeivel. Válasszunk az $\{1, 2, \ldots, \Delta + 1\}$ halmazból uniform módon egy elemet. Ha ez az $\{1, 2, \ldots, d(v)\}$ halmazba esik akkor v-ből a megfelelő szomszédba lépünk át. Különben maradunk a v csúcsban. A helyben maradásnak mindig pozitív valószínúsége van. (Emiatt nem lesz problémánk a páros gráfok esetével.)

Kezdőpontunk eloszlását egy $\vec{k}$ eloszlásvektor (nemnegatív vektor, komponenseinke összege 1) írja le (vektoraink, soraink, oszlopaink pozícióit a V csúcshalmazzal azonosítottuk). A mi véletlen sétánknál ez különösen egyszerű: s-ben 1 komponensű, minden más komponens értéke 0. A következő észrevétel más kiinduló eloszlásnál és más átmeneti mátrixnál is igaz.

5. Lemma. *A véletlen sétánk L lépés utáni helyzetének eloszlása $M^L\vec{k}$.*

A lemma csupán a definíciók és a mátrix aritmetika ismeretét kívánja. Az olvasóra hagyjuk.

M szimmetrikus, így sajátértékei valósak ($\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, ahol $n = |V|$) és alkalmas Q ortogonális mátrixszal

$$M = Q \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{pmatrix} Q^{-1} = Q\Lambda Q^{-1}.$$

A következő lemmából

6. Lemma. *M sajátértékeinek abszolútértéke legfeljebb 1. Az 1 sajátérték, -1 pedig nem sajátérték. Ha G összefüggő, akkor az 1 sajátérték multiplicitása 1.*

Valóban: Legyen λ egy sajátérték és $\vec{v}$ egy hozzátartozó sajátvektor. Vegyük a $\vec{v}$ sajátvektor egy maximális abszolútértékű komponensét. Mi lesz ezen komponens éréke $M\vec{v}$-ben? Egyrészt M sztochasztikus mátrix, így minden komponense $\vec{v}$ komponenseiből átlagolódik ki, abszolútértékben nem nőhet. Másrészt λ-szorosa lesz. Így valóban $|\lambda| \leq 1$. A csupa 1 vektor (továbbiakban $\vec{j}$) sajátvektor 1 sajátértékkel. Sőt az 1 sajátértékhez tartozó összes sajátvektor konstans komponensű vektor

($\vec{j}$ számszorosa): Legyen $\vec{s}$ egy tetszőleges sajátvektor 1 sajátértékkel. Mint előbb most is vegyük a $\vec{s}$ sajátvektor egy maximális abszolútértékű komponensét, értékét jelöljük C-vel. Ekkor ezen komponense az M mátrix-szal való szorzás után a saját és a szomszédokhoz tartozó komponensek értékéből átlagolódik ki, közben nem változik (egyszerese lesz önmagának). Ez csak úgy lehet, ha minden szomszédjához tartozó komponens értéke is C volt. A gondolatmenet ismételgetésével, felhasználva, hogy G összefüggő kapjuk, hogy $\vec{s} = C \cdot \vec{j}$. Ha feltennénk, hogy -1 sajátérték és megismételnénk a fenti gondolatmenetet egy hozzátartozó sajátvektorral, akkor ellentmondásra jutnánk: az átlagolásnál a megfelelő komponens is szerepet játszik (ezért tértünk át $\widetilde{A}$-ra), a -1-szereződés nem lehetséges.

A lemmából adódik

$$M^L = Q\Lambda^L Q^{-1} = Q \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \lambda_2^L & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n^L \end{pmatrix} Q^{-1},$$

ami egy konvergens mátrixsorozat ($\max_{i=2,3,\ldots,n} |\lambda_i| < 1$).

Könnyű látni, hogy a határeloszlás stacionárius is egyben (M-mel szorozva nem változik), azaz az 1 sajátértékhez tartozó sajátvektor számszorosa. Tehát a határeloszlás az uniform eloszlás.

A bizonyítás hátralévő része a Markov-láncok elméletére támaszkodik. Csak megemlítjük az alábbi lemmát, ami összegzi a szükséges bizonyítandó állításokat.

Legyen $\rho_t(u)$ az a valószínűségi változó, amely megmondja a véletlen séta folyamán az első $t-1$ lépéssel meglátogatott t csúcs között milyen arányban szerepel a u csúcs. Legyen uv egy él a gráfunkban, ekkor $\nu(u)$ az a lépésszám, amit egy u pontból indulva megteszünk a következő u látogatásig. Legyen uv egy él a gráfunkban, ekkor $\sigma(uv)$ az a lépésszám, amit egy uv lépés után teszünk a következő uv (sorrend számít) lépéssel bezárólag.

7. Lemma. *(i)* $\mathbb{E}(\rho_t(u)) \to \frac{1}{|V|}$,

(ii) $\mathbb{D}(\rho_t(u)) \to 0$,

(iii) $\mathbb{E}(\nu(u)) = |V|$,

(iv) $\mathbb{E}(\sigma(uv)) = (\Delta + 1)|V| \leq |V|^2$.

A lemma alapján egyszerűen becsülhető, hogy mi azon séta hosszának várható értéke, amely az s csúcsból egy t-be vezető legrövidebb séta éleinek megfelelő lépéseket sorban (közbeiktatott lépésekkel) megteszi. ■

Megjegyzés. Az algoritmus minden probléma nélkül kiterjeszthető irányított gráfokra. Hogy ez az algoritmus hasznos legyen szükségünk lenne véletlen séta fenti analízisére irányított gráfokra. Sajnos a megfelelő tétel nem igaz. Ehhez vegyünk egy gráfot, amiben s-et és t-t egy n pontú út köti össze. az út minden pontjából vezessen egy irányított él s-be is. Azaz sétánk során minden t-től különböző csúcsban két lehetőségünk van: vagy t-felé lépünk, vagy visszatérünk a kiinduló pontba. Így a véletlen séta akkor éri el t-t, ha $n-1$-szer egy két kimenetelű véletlen választásból mindig a kedvező következik be. Erre várhatóan exponenciális sokáig kell várnunk. Az ELÉRHETŐSÉG probléma (amiről tudjuk, hogy $\mathcal{NL}$-teljes) $\mathcal{RL}$ osztályba esése meglepő lenne.

A fenti klasszikus tétel kutatások hosszú irányát indította el. A véletlen séták, az úgy nevezetett expander gráfok, a derandomizálási módszerek kutatásának egyik kiemelkedő eredménye a következő tétel, amit bizonyítás nélkül említünk meg.

8. Tétel (Reingold, 2008). *IRÁNYÍTATLAN-ELÉRHETŐSÉG*$\in \mathcal{L}$.