11. cvičení

Využití náhody v různých oblastech teoretické informatiky

Úloha 1 (Elektřina?)
Začneme poměrně překvapivou souvislostí náhodných procházek po grafu a fyziky. Mějme graf $G = (V, E)$ se dvěma význačnými vrcholy $u \neq v$. Na tomto grafu provedeme náhodnou procházku: tedy začneme v nějakém vrcholu w, a poté se v každém kroku přesuneme do uniformně náhodně vybraného souseda vrcholu, v němž právě jsme (doteď jsme u, v nepotřebovali).[1]
Ale nás taky může zajímat, jaká je pravděpodobnost, že se z nějakého vrcholu dostaneme dříve do u než do v. Pro libovolný vrchol x zadefinujeme $p_x := P[\text{náhodná procházka z } x \text{ dojde dříve do } u \text{ než do } v]$. Berme jako fakt, že tato pravděpodobnost je dobře definovaná, a snadno si všimneme následujících vlastností: $p_u = 1, p_v = 0, \forall x \in V \setminus \{u, v\} : p_x = \sum_{y \in N(x)} \frac{p_y}{\deg(x)}$.
Uvažme dále elektrický obvod se stejnou strukturou jako G - tedy vrcholy odpovídají uzlům, hrany odpovídají rezistorům. Představme si, že každá hrana má odpor 1 ohm, a mezi vrcholy u, v zapojíme jednovoltový zdroj (tak, aby napětí v u bylo 1, a napětí ve v bylo 0). Fyzikální pravidla nám říkají, že v uzlu (vrcholu) x je nějaký odpor U_x. Také každou hranou protéká proud $I_{x,y} := U_x - U_y$ – ten může být záporný, záleží, kde je vyšší napětí (správně bychom měli dělit odporem hrany, ale odpory jsme chytře nastavili jako jednotkové, takže všechno funguje v pořádku). Platí nám také Kirchhoffův zákon: pro libovolný vrchol $x \in V \setminus \{u, v\} : \sum_{y \in N(x)} I_{x,y} = 0$.

- Ukažte, že U_x a p_x jsou definované stejnými rovnicemi.

Poznamenejme, že tohle je celkem užitečný koncept pro studování náhodných procházek na grafech. Přirozeně jde definovat koncept efektivního odporu $R^{\text{eff}}_{u,v} = \frac{U_u - U_v}{\sum_{y \in N(u)}} I_{u,y}$ (rozdíl napětí dělený opravdovým proudem vycházejícím z u). Jednak jde pak poměrně snadno dokázat fyzikální vlastnost Rayleighova principu monotonie[2] čistě matematicky, aniž bychom se museli snažit dělat nějaké experimenty. Ale zároveň máme následující větu: označíme-li si $T_{u,v,u}$ délku cesty z u do v a pak zpět do u (tohle je náhodná veličina!), platí, že $\mathbb{E}[T_{u,v,u}] = 2 \cdot |E(G)| \cdot R^{\text{eff}}_{u,v}$.
Toto dále použijeme, abychom ukázali, že v lízátku existují dva vrcholy, mezi nimiž aspoň jedním směrem je potřeba ve střední hodnotě kubicky mnoho kroků ve velikosti grafu. Graf lízátko (označme jej třeba L_n) je graf na n vrcholech, kde prvních $n/2$ vrcholů tvoří kliku, a druhých $n/2$ vrcholů tvoři cestu, která je připojena k právě jednomu vrcholu kliky. Důkaz proveďte následovně:

a) Zvolte vhodné dva vrcholy; hint: *Zkuste vrcholy, co jsou od sebe co nejdál.*

b) Prvním krokem je odhadnout zdola efektivní odpor mezi těmito vrcholy: je potřeba si rozmyslet, jaký je efektivní odpor cesty na ℓ vrcholech, a dále, co se s efektivním odporem stane, když zapojíte za sebe dva grafy sériově. Tohle si raději nechte až na konec a použijte jako fakt odhad $\geq n/2$.

c) Odhadněte zdola počet hran grafu.

d) Součin těchto dvou odhadů nám dává dolní odhad na střední hodnotu $T_{u,v,u}$.

Jak poslední fakt zde zmiňme větu, která říká, že asymptoticky horší situace než kubická nemůže nastat: Gregory Lawler dokázal, že pro každý souvislý neorientovaný graf na n vrcholech a libovolné dva vrcholy u, v platí, že střední hodnota doby návštěvy vrcholu v po vyjití z u může být nejvýše kubická.

Úloha 2 (Pravděpodobnostní algoritmus pro dosažitelnost)
V předchozí úloze jsme si řekli, že když v n-vrcholovém souvislém neorientovaném grafu vyjdeme z vrcholu u, střední hodnota doby než navštívíme v, je nejvýše kubická. Konkrétně ji můžeme odhadnout shora jako $\leq 2n^3$.

[1] Pokud chcete formální definici náhodné procházky, můžeme použít například následující: náhodná procházka na grafu $G = (V, E)$ je posloupnost náhodných veličin $X_0, X_1, \ldots$, kde $P[X_0 = w] = 1$, $\forall z \in V \setminus \{w\} : P[X_0 = z] = 0$, a $\forall i \geq 1 \forall x, y \in V : P[X_i = x | X_{i-1} = y] = \begin{cases} \frac{1}{\deg(y)} & \text{pokud } \{x, y\} \in E \\ 0 & \text{jinak} \end{cases}$.

[2] Zhruba: Když vezmeme nějaký obvod, a snížíme odpor nějakého rezistoru, pak efektivní odpor nemůže stoupnout.

Sestrojte pravděpodobnostní algoritmus[3] pro rozhodnutí dosažitelnosti (tedy odpověď na otázku „existuje cesta z u do v?“), který selže s pravděpodobností nejvýše $1/4$, a potřebuje jenom logaritmický prostor.
Technická poznámka k logaritmickému prostoru: když máme algoritmus běžící v sublineárním prostoru, počítáme jej tak, že máme vstup na „read-only“ disku, kde k němu můžeme přistupovat, ale nemůžeme jej upravovat (ale podle potřeby si jej můžeme bez problému kopírovat).[4]
Hinty: *Když cesta nebude existovat, náš algoritmus určitě odpoví korektně.*
Může se hodit Markov.

Co se asi tak vejde do logaritmického prostoru?
Opět jen pro zajímavost: je celkem snadné ukázat, že dosažitelnost je řešitelná v nedeterministickém logaritmickém prostoru (což odpovídá třídě NL, což je ekvivalent NP s logaritmickým prostorem), a Szelepcsényi-Immerman ukázali, že dosažitelnost je NL-těžká[5], a je v coNL[6], což tedy ukazuje, že třídy NL a coNL jsou stejné.
My jsme teď zároveň ukázali, že dosažitelnost umíme vyřešit v logaritmickém prostoru pravděpodobnostním algoritmem. Na druhou stranu je stále otevřenou otázkou, zda lze dosažitelnost vyřešit v logaritmickém prostoru *deterministickým* algoritmem.

Úloha 3 (Pravděpodobnostní algoritmus pro k-SAT)
Problém 3-SAT je následující: na vstupu máme formuli, která je v 3-CNF (konjunkce disjunkcí, kde v každé disjunkci je nejvýše 3 literálů) na n proměnných – například $(x_1 \vee x_2) \wedge (x_2 \vee \neg x_3 \vee \neg x_4) \wedge (\neg x_1 \vee x_4)$.
Uvažme následující algoritmus:

- Na začátku všechny proměnné nezávisle uniformně náhodně ohodnoť.
- Dokud formule není splněná naším ohodnocením, opakuj následující proceduru (ale nejvýše $n/2$-krát): vezmeme jednu nesplněnou klauzuli, a jednu proměnnou v ní přehodíme tak, aby tato klauzule byla splněná.
- Pokud se nám takto podařilo formuli splnit, řekneme, že je splnitelná, jinak řekneme, že je nesplnitelná.

Proveďte analýzu tohoto algoritmu (pro jednoduchost klidně předpokládejte, že n je sudé):

- Jak to je s případem, kdy je formule nesplnitelná?
- Uvažme nyní případ, kdy je formule splnitelná, a uvažme jedno konkrétní splňující ohodnocení a.
- Pravděpodobnost toho, že algoritmus správně řekne, že je formule splnitelná budeme odhadovat pomocí našeho zafixovaného a: nejdříve se musíme trefit aspoň polovinou proměnných stejně jako a, a pokud se trefíme, tak se přinejhorším musíme trefit správně $n/2$-krát, takže potom se musíme přinejhorším $n/2$-krát trefit v nesplněné klauzuli tak, že proměnnou, která byla nastavena špatně, nastavíme na správně. Otázka, která zbývá na vás je: nalezněte konstantní dolní odhad pravděpodobnosti, že se při náhodném splnění jedné nesplněné klauzule trefíme tak, abychom proměnnou, jejíž hodnotu přepneme, změnili na správnou.
- Tento algoritmus evidentně běží v polynomiálním čase, ale má dost velkou chybovost. Co se stane, když algoritmus pustíme vícekrát za sebou? Zkuste najít takový počet spuštění, aby časová složitost algoritmu byla $\mathrm{poly}(n) \cdot 3^{n/2}$ a pravděpodobnost chyby byla nejvýše $1/n$. Může se vám hodit nerovnost $1 + x \leq e^x$ (či spíše ve formě $1 - x \leq e^{-x}$).
- Co kdybychom místo 3-SATu řešili obecný k-SAT pro $k \geq 4$? Jak to dopadne při porovnání s naivním algoritmem?

[3] Pravděpodobnostní algoritmy jste asi už potkali, ale zkráceně: bude se jednat o algoritmus, který se bude moct v každém kroku rozhodnout náhodně z několika možností

[4] Technicky tuto třídu definujeme na Turingových strojích, kde máme read-only vstupní pásku, a druhou pracovní pásku. Do prostorové složitosti se nám počítá jenom pracovní páska.

[5] Pozor, těžkost pro NL bereme vzhledem k redukci v logaritmickém prostoru!

[6] Při „konedeterminismu“ nehledáme jednoho svědka, díky kterému bychom mohli ověřit řešení v logaritmickém prostoru, ale naopak chceme, aby neexistoval svědek, který řešení v logaritmickém prostoru vyvrací. Hezký příklad pro rozdíl mezi NP a coNP je rozdíl mezi SAT a TAUT: v SATu dostaneme formuli, a máme rozhodnout, jestli je splnitelná – pak nám stačí jako svědek jedno splňující ohodnocení, které ukazuje splnitelnost. V TAUT dostaneme formuli, a máme rozhodnout, zda se jedná o tautologii, tedy vždy splněnou formuli – pak máme snadno svědka, když tato formule není vždy splnitelná: *nesplňující ohodnocení*.