A számítástudomány alapjai 2021. I. félév

12. gyakorlat. Összeállította: Fleiner Tamás (fleiner@cs.bme.hu)

Tudnivalók

Def: Az algoritmus fogalmát nem definiáljuk, de algoritmusra úgy gondolunk, mint valamiféle elméleti számítógépen futtatott programra, mely egy, az input által megadott konkrét feladatot old meg, azaz véges sok lépés megtétele után a helyes megoldás lesz az adott inputhoz tartozó output. Ez a konkrét feladat mindig egy általános Π probléma egy, az input által megadott konkrét esete. Természetesen ugyanazt a Π problémát több különböző algoritmus is megoldhatja. Ha tehát A egy ilyen algoritmus a Π problémára, akkor A *inputja* (bemenete) jelöli ki, hogy a szóban forgó Π problémacsoportnak pontosan melyik konkrét feladatáról is van szó. Ezen input *mérete* pedig az inputot leíró bitek száma. Ennek megfelelően tetszőleges Π problémát megoldó A algoritmushoz tartozik egy f_A függvény: $f_A(n)$ azt adja meg, hogy legfeljebb hány lépést tesz meg az A algoritmus a legfeljebb n hosszúságú inputokon. A lépésszámfüggvény szempontjából tehát lényegtelen, ha az A algoritmus a legfeljebb n hosszúságú inputok $99,999\%$-án néhány lépésben végez, $f_A(n)$ a legrosszabbul viselkedő, legfeljebb n hosszúságú inputhoz tartozó eset lépésszáma. A lépésszámra vonatkozó felső becslés ezért garantált felső korlátot ad a legfeljebb n hosszúságú inputtal futtatott program lépésszámára. Mi a továbbiakban olyan algoritmusokat próbálunk keresni a vizsgált problémákra, amelyek minél jobb (azaz minél kisebb) lépésszámgaranciával rendelkeznek.

Példa: Ha az algoritmus inputja egy pozitív egész n szám, akkor az input mérete az n bináris alakjában található jegyek száma, azaz $1 + \lfloor \log_2(n) \rfloor$. Ha az input egy n csúcsú egyszerű $G = (V, E)$ gráf, akkor G-t a szomszédossági mátrixával megadva az input mérete n^2. (Vannak persze értelmesebb megadások is, sőt, szinte csak azok vannak. Éllistával pl. az input mérete $konst \cdot (n+m)$, ahol m a G éleinek száma.)

Def: Az A algoritmus *polinomidejű* (néha: *polinomiális* vagy *hatékony*), ha létezik olyan $p(n)$ polinom, amelyre $f_A(n) \leq p(n)$ teljesül minden $n \geq 1$-re. Az A algoritmus *exponenciális idejű*, ha létezik olyan pozitív K és $c > 1$ konstansok, melyekre $f_A(n) \leq K \cdot c^n$ teljesül minden $n \geq 1$-re.

Megjegyzés: Ha a számítógép műveleti sebessége (mondjuk) kétszeresére gyorsul, akkor egy polinomidejű algortimussal egységnyi idő alatt egy konstansszor nagyobb inputméretű problémát oldhatunk meg, mint a régi számítógéppel, míg exponenciális futásídeű algoritmus esetén csak konstanssal növekszik az egységnyi idő alatt megoldható input mérete.
Itt és most a polinomidejű algoritmust hatékonynak tekintjük, az olyat pedig, nem szeretjük, amire az exponenciális becslésnél nem tudunk jobbat mondani. (Minden polinomidejű algoritmus természetesen exponenciális idejű is (hisz exponenciális felső becslés is adható a futásidőre), de az exponenciális idejű algoritmusok nem feltétlenül polinomidejűek.

Példa: A BFS algoritmus hatékony, hiszen egy n élű m csúcsú gráfot (amelyet meg lehet adni n^2 méretű vagy $konst \cdot (n+m)$ méretű inputtal) a lépésszámra $f_{BFS}(n) \leq c \cdot (n+m)$ teljesül alkalmas c konstansra, tehát $p(n) = c' \cdot n$ megfelelő polinom, ahol c' alkalmas konstans.

Def: *Döntési probléma* az olyan probléma, amelyek outputja egyetlen bit, azaz minden értelmes inputhoz az „IGEN" vagy a „NEM" outputok valamelyike tartozik. A P problémaosztályt mindazon döntési problémák alkotják, amelyekre van polinomidejű algoritmus.

Def: NP-beli probléma alatt olyan Π döntési problémát értünk, amelyhez minden olyan inputra, amelyre „IGEN" az output, ez a tény polinomidőben bizonyítható. Formálisan: $\Pi \in NP$, ha létezik egy A algoritmus az alábbi tulajdonságokkal. (1) Az A algoritmus inputja tetszőleges I, T pár, és az A algoritmus lépésszáma felülről becsülhető I méretének polinomjával. (Tehát T-től nem függ.) (2) A Π minden olyan I inputjához, amelyhez „IGEN" a helyes output, létezik olyan T tanú, hogy az A algoritmust az (I, T) inputtal elindítva „IGEN" választ ad, valamint (3) minden olyan I inputhoz, amelyhez „NEM" a helyes output, A az (I, T) inputon minden T-re „NEM" választ ad.
A $co-NP$-beli problémaosztályt azon Π döntési problémák alkotják, amelyekhez létezik minden „NEM" válaszra polinomidőben ellenőrizhető tanú.

Megfigyelés: $P \subseteq NP \cap co-NP$.

Def: A Π probléma *polinomidőben visszavezethető* a Π' problémára (jelölése $\Pi \preceq \Pi'$), ha a Π tetszőleges I inputjához polinomidejű algoritmussal konstruálható a Π' problémának olyan I' inputja, melyre (Π'-ben) ugyanaz a válasz, mint I-re Π-ben.

Megfigyelés: (1) $\Pi \preceq \Pi' \preceq \Pi'' \Rightarrow \Pi \preceq \Pi''$. (2) $\Pi \preceq \Pi' \in P \Rightarrow \Pi \in P$.

Def: A Π döntési probléma NP*-nehéz*, ha $\Pi' \preceq \Pi$ minden $\Pi' \in NP$ esetén, azaz minden NP-beli probléma visszavezethető Π-re. A Π NP*-teljes*, ha $\Pi \in NP$ és Π NP-nehéz.

Megfigyelés: Ha Π NP-teljes és $\Pi \preceq \Pi' \in NP$, akkor Π' is NP-teljes.

Def: A SAT probléma inputja egy CNF (konjunktív normálforma), outputja „IGEN", ha az inputban szereplő logikai változók logikai értéke megválasztható úgy, hogy az adott CNF kiértékelése igaz legyen. Minden CNF klózok összeéselése, minden klóz literálok összevagyolása és minden literál azonos valamelyik logikai változóval vagy annak negáltjával.

Példa: : $(\overline{x}_1 \vee x_3 \vee \overline{x}_7) \wedge (x_2 \vee x_6) \wedge (x_2 \vee \overline{x}_4 \vee x_5 \vee x_7) \wedge (\overline{x}_4 \vee \overline{x}_6)$

Cook-Levin-tétel: A SAT probléma NP-teljes.

A Cook-Levin-tétel miatt a világ kétféle lehet. Vagy $P = NP$, és ekkor minden NP-beli probélmára van polinomidejű algoritmus. Van tehát P-beli NP-teljes probléma is (valójában minden P-beli probléma NP-teljes). Sőt: elég egyetlenegy NP-teljes problémát találni P-ben, mert ebből következik, hogy $P = NP$. Vagy $P \neq NP$, és ekkor minden NP-teljes probléma reménytelen abban az értelemben, hogy bizonyosan nincs rá polinomidejű algoritmus. Ez utóbbiban hisz a többség.

További NP-teljes problémák: (1) HAM (Input: G gráf, output: „IGEN", ha G-nek van Hamilton köre), (2) 3-SAT (a SAT speicális esete, ahol minden klóz pontosan 3 db literált tartalmaz), (3) 3-SZÍN (Input: G gráf, output: „IGEN", ha $\chi(G) \leq 3$), (4) $k \geq 3$ esetén a k-SZÍN (Input: G gráf, output: „IGEN", ha $\chi(G) \leq k$), (5) MAXFTN (Input: G gráf, $k > 0$ egész, output: „IGEN", ha $\alpha(G) \geq k$) (6) MAXKLIKK (Input: G gráf, $k > 0$ egész, output: „IGEN", ha $\omega(G) \geq k$)

Gyakorlatok

1. Tegyük fel, hogy A egy polinomidejű algoritmus a Π problémára. Legyen Π' egy másik probléma, és legyen A' olyan polinomidejű algoritmus, amely Π' tetszőleges I' inputjához a Π olyan I inputját készíti el, amelyhez a Π problémában ugyanaz a válasz tartozik, mint I'-höz a Π' problémában. Helyes és polinomidejű-e az az A^* algoritmus a Π' problémára, amelyet úgy kapunk, hogy Π' tetszőleges I' inputján lefuttatjuk az A' algotitmust, majd a kapott I outputot inputként felhasználva lefuttatjuk az A algoritmust?
2. Tegyük fel, hogy az A algoritmus a Π problémát oldja meg olyan módon, hogy Π tetszőleges n méretű inputjához A n lépésben elkészíti a Π probléma egy $\lceil n/2 \rceil$ és egy $\lfloor n/2 \rfloor$ méretű inputját, és azokat megoldja saját maga meghívásával. Polinomidejű-e az A algoritmus? Mi a helyzet akkor, ha tetszőleges n méretű inputból a két elkészített input mérete $n - 10$?
3. Mutassuk meg, hogy az alábbi problémák P-beliek:
   (a) Adott G irányítatlan gráfról döntsük el, van-e benne kör.
   (b) Adott G irányítatlan gráfról és k pozitív egészről döntsük el, van-e olyan részgráfja, amiben minden fok $\geq k$.
   (c) Adott G irányítatlan gráfról döntsük el, van-e K_{10} részgráfja.
   (d) Adott G öf gráfról és $c : E(G) \to \mathbb{R}_+$ élsúlyokról döntsük el, igaz-e,hogy G bármely feszítőfájának a költsége legalább k.
   (e) 2-SAT (A SAT probléma, ahol minden klóz legfeljebb két literált tartalmaz.)
4. Mutassuk meg, hogy az alábbi problémák NP-beliek:
   (a) Adott G irányítatlan gráfról és k pozitív egészről döntsük el, van-e k-reguláris részgráfja.
   (b) Adott G öf gráfról és $c : E(G) \to \mathbb{R}_+$ élsúlyokról döntsük el, igaz-e, hogy G-nek van pontosan k költségű feszítőfája.
   (c) Adott G gráfról és $l : E(G) \to \mathbb{R}$ (esetleg negatív) élhosszokról döntsük el, igaz-e, hogy G bármely két csúcsának a távolsága legfeljebb k
5. Mutassuk meg, hogy az alábbi problémák $co - NP$-beliek:
   (a) Adott G gráfról döntsük el, síkbarajzolható-e.
   (b) Adott $2n$ csúcsú G gráfról döntsük el, igaz-e, hogy bármely n csúcsa páros gráfot feszít.
   (c) Adott G gráfról döntsük el, igaz-e, hogy $\omega(G) \leq k$.
   (d) Adott n számról döntsük el, hogy prímhatvány-e.
6. Mutassuk meg, hogy az alábbi problémák $NP \cap co - NP$-beliek:
   (a) Adott G gráfról döntsük el, páros-e.
   (b) Adott G gráfról döntsük el, összefüggő-e.
   (c) Adott G páros gráfról döntsük el, van-e teljes párosítása.
   (d) Adott hálózatról döntsük el, van-e benne k nagyságú folyam.
   (e) Adott n és k egészekről döntsük el, relatív prímek-e.
7. Mutassuk meg, hogy az alábbi problémák NP-teljesek:

(a) FÉLHAM inputja egy G gráf, outputja „IGEN”, ha G-nek van olyan köre, ami G csúcsainak legalább a felét tartalmazza

(b) FELE-3-SZÍN inputja egy G gráf, outputja „IGEN”, ha G-nek van olyan 3-színezhető feszített részgráfja, amely G csúcsainak legalább a felét tartalmazza.

(c) MAXTÁV inputja egy $G = (V, E)$ gráf, egy $l : E \to \mathbb{R}_+$ hosszfüggvény valamint egy $k \in \mathbb{R}_+$ szám. Az output akkor „IGEN”, ha G-ben van legalább k összhosszú út.

8. Az alábbi döntési problémák mindegyikéről döntsük el, hogy a $P, NP, co-NP, NP-teljes$ osztályok melyikébe tartozik. Input egy G irányítatlan gráf, Output „IGEN”, ha
(a) van G-ben 99 pontú klikk, (b)ha G-ben van 5 pontú kör, (c) ha G-ben van legalább $\sqrt{n}$ pontú kör, (d) ha G élei pirosra és zöldre színezhetők úgy, hogy ne legyen egyszínű ptn kör. (*)