NP

Kakve algoritme želimo?

Želimo algoritme koji su rešivi u polinomijalnom vremenu.

Zbog čega nam algoritmi eksponencijalne složenosti ne odgovaraju?

(pretpostavimo da imamo računar sa brojem procesora koliko ima elektrona u kosmosu, svaki procesor ima jačinu trenutno najjačeg super-kompjutera, i pustimo ga da radi celo vreme postojanja kosmosa, neće rešiti problem putničkog trgovca za 1000 gradova)

Koji su to onda problemi polinomijalne složenosti (a problemi rešivi)?

Primeri problema

LSOLVE. Dat je sistem linearnih jednačina, naći rešenje:

$$\begin{array}{l} 0x_0 + 1x_1 + 1x_2 = 4 \\ 2x_0 + 4x_1 - 2x_2 = 2 \\ 0x_0 + 3x_1 + 15x_2 = 36 \end{array} \qquad \begin{array}{l} x_0 = -1 \\ x_1 = 2 \\ x_2 = 2 \end{array}$$

LP. Dat je sistem linearnih nejednačina, naći rešenje:

$$\begin{array}{l} 48x_0 + 16x_1 + 119x_2 \leq 88 \\ 5x_0 + 4x_1 + 35x_2 \geq 13 \\ 15x_0 + 4x_1 + 20x_2 \geq 23 \end{array} \qquad \begin{array}{l} x_0 = 1 \\ x_1 = 1 \\ x_2 = 1/5 \end{array}$$

ILP. Dat je sistem linearnih nejednačina, naći 0-1 rešenje:

$$\begin{array}{l} x_1 + x_2 \geq 1 \\ x_0 + x_2 \geq 1 \\ x_0 + x_1 + x_2 \leq 2 \end{array} \qquad \begin{array}{l} x_0 = 0 \\ x_1 = 1 \\ x_2 = 1 \end{array}$$

SAT. Zadovoljivost izraza iskazne logike sa `n` promenljivih.

$$(\neg x_1 \vee \neg x_2) \wedge (x_0 \vee x_2) \qquad x_1 = \bot,\ x_0 = \top,\ x_2 = \bot$$

Koji od ovih problema su rešivi (u polinomijalnom vremenu)?

- **LSOLVE**. Jeste, Gausovom eliminacijom;
- **LP**. Jeste, elipsoidnim algoritmom (pitanje je bilo decenijama otvoreno);
- **ILP, SAT**. Nije otkriven polinomijalni algoritam (i, veruje se da ne postoji).

P - problemi za koje postoje algoritmi polinomijalne složenosti

Problemi pretrage

Pokušavamo da svedemo probleme na problem pretrage. (ili, alternativno, na probleme za koje je posle izvršavanja moguće odgovoriti sa "da", odnosno "ne")

- **Problem pretrage:** Za instancu problema `I`, naći rešenje `S`.
- **Zahtev:** Provera da li je dato `S` rešenje mora biti polinomijalne složenosti.

LSOLVE. Provera da li su dati brojevi x_i rešenje sistema linearnih jednačina se svodi na zamenu promenljivih u sistemu jednačina njihovim vrednostima i proveri jednakosti.

ILP. Provera je slična kao i za LSOLVE.

FACTOR. Naći (netrivijalni) delitelj datog broja `n`. Provera je trivijalna - dovoljno je podeliti `n` datim rešenjem `s` i proveriti da li je ostatak nula.

P i NP

P - Klasa problema za koje postoje algoritmi polinomijalne složenosti (LSOLVE, LP, SORT, put između dva čvora grafa)

NP (Nondeterministic Polynomial) - Klasa problema pretrage (klasična definicija, kao u knjizi, ograničava ovo na "da-ne" probleme)

Da li su sledeći problemi NP?

- **FACTOR**: Naći (netrivijalni) delitelj datog broja `n`;
- **LP**: Dat je sistem linearnih nejednačina, naći rešenje;
- **TSP**: Dat je težinski graf `G`, naći najkraći prost ciklus koji obilazi svaki čvor tačno jednom;
- **HAMILTON-CYCLE**: dat je neusmereni graf, naći prost ciklus koji obilazi svaki čvor tačno jednom;

SAT

Dat je bulovski izraz. Naći rešenje (ili, naći da li rešenje postoji - da li je izraz zadovoljiv).

Upotreba

- Dizajn hardvera

- Automatska verifikacija softverskih sistema
- ...

Ne znamo efikasan algoritam. Pretpostavljamo da problem nije rešiv.

Teorema: Ako je SAT moguće polinomijalno svesti na neki problem Y, zaključujemo da Y nije rešiv (uz pretp. da SAT nije rešiv).

Primer: Svođenje SAT-a na ILP.

Bilo koji SAT problem je moguće svesti na oblik:

$$
\begin{aligned}
\neg x_1 \vee\ \ x_2 \vee\ \ x_3 &= T \\
\ \ x_1 \vee \neg x_2 \vee\ \ x_3 &= T \\
\neg x_1 \vee \neg x_2 \vee \neg x_3 &= T \\
\neg x_1 \vee \neg x_2 \vee\ \ x_4 &= T
\end{aligned}
$$

Svodimo na ILP:

$$
\begin{aligned}
1 &\leq (1 - x_1) + x_2 + x_3 \\
1 &\leq x_1 + (1 - x_2)\ \ + x_3 \\
1 &\leq (1 - x_1) + (1 - x_2) + (1 - x_3) \\
1 &\leq (1 - x_1) + (1 - x_2) + x_4
\end{aligned}
$$

Dakle, ILP je bar onoliko težak koliko je to SAT. To jest, ako bi ILP bio rešiv, i SAT bi bio rešiv.

NP-kompletnost

Problem je NP-težak ako se svaki problem iz NP može polinomijalno redukovati na njega.

Problem X je NP-kompletan ako je NP-težak i $X \in NP$.

Teorema (Kuk) SAT je NP-kompletan.

Model izračunavanja

Deterministička Tjuringova mašina. Sastoji se od upravljačkog bloka sa konačnim brojem stanja, glave za čitanje i pisanje i trake koja je beskonačna sa obe strane.

Vreme posmatramo kao broj koraka do zaustavljanja DTM. Ako se program M zaustavlja za sve svoje ulaze, onda definišemo vremensku složenost kao:

```
TM(n) = max { m | postoji ulaz x, |x| = n, tako da je
                  vreme izvršavanja programa M za ulaz
                  x jednako m }
```

M je polinomijalni program za DTM ako postoji polinom p takav da je

```
(∀n ∈ Z⁺) TM(n) ≤ p(n).
```

Klasu problema P definišemo kao

```
P = { L | postoji polinomijalni program M za DTM tako da je LM=L }
```

Polinomijalna proverljivost karakteriše klasu NP. Dakle, da li je za dato rešenje problema moguće u polinomijalnom vremenu na DTM proveriti da li je stvarno rešenje.

Nerešivost

Ponekad ima prednosti:

- enkripcija (faktorizacija brojeva naspram množenja)

Ponekad postoje rešenja za specijalne instance problema:

- postoji rešenje linearne složenosti za 2-SAT (maksimum dve prom. po jednačini)
- postoji rešenje linearne složenosti za Horn-SAT (maksimum jedna ne-negirana prom. po jednačini)

Zadaci

Zadatak 1

Pokazati da su svaka dva netrivijalna problema iz klase P polinomijalno ekvivalentna. (netrivijalan problem - problem za koji postoje ulazi koji vraćaju "da" i ulazi koji vraćaju "ne")

Rešenje:

Neka su B i C netrivijalni problemi iz P. Neka je u_0 ulaz za problem C za koji je rešenje "ne", a u_1 ulaz za koji je rešenje "da" ($u_0 \notin Lc$, $u_1 \in Lc$).

Za svođenje B na C možemo koristiti sledeći polinomijalni algortiam:

- ulazu v za B pridružujemo $f(v) = u_0$, ako $v \notin LB$
- ulazu v za B pridružujemo $f(v) = u_1$, ako $v \in LB$

Tada važi: $v \in LB$ akko $f(v) \in LC$, to jest, B je polinomijalno svodljiv na C.

Analogno, suprotan smer.

Zadatak 2

Dokazati da je problem 3SAT NP-kompletan.

3SAT - dat je bulovski izraz u KNF gde svaka klauza sadrži tačno 3 promenljive. Treba ispitati da li je zadovoljiv.

Rešenje:

$3SAT \in NP$ jer je za dato rešenje trivijalno proveriti da li zadovoljava izraz (za polinomijalno vreme)

Sledeći korak je da dokažemo da se proizvoljan SAT problem može svesti na neki 3SAT problem. Proizvoljan ulaz za SAT svodimo na ulaz za 3SAT tako da je rešenje problema SAT "DA" AKKO je "DA" rešenje problema 3SAT.

Neka je $C = (x_1 + x_2 + ... + x_k)$ proizvoljna klauza. Dodajemo promenljive $y_1, ..., y_{k-3}$ i definišemo klauzu C':

$$C' = (x_1 + x_2 + y_1)(x_3 + \neg y_1 + y_2)(x_4 + \neg y_2 + y_3)... \\ (x_{k-2} + \neg y_{k-4} + y_{k-3})(x_{k-1} + x_k + \neg y_{k-3})$$

C' je zadovoljiva akko je C zadovoljiva (dokazati). Takođe, izraz E' (dobijen zamenom klauze C u izrazu E klauzom C') je zadovoljiv akko je E zadovoljiv.

Klauzu $(x_1 + x_2)$ menjamo sa $(x_1 + x_2 + z)(x_1 + x_2 + \neg z)$, a (x_1) sa $(x_1 + y + z)(x_1 + \neg y + z)(x_1 + y + \neg z)(x_1 + \neg y + \neg z)$.

Zadatak 3

Neka je E izraz u KNF u kome se svaka promenljiva x pojavljuje najviše jednom i njena negacija se javlja najviše jednom.

Konstruisati algoritam polinomijalne vremenske složenosti za utvrđivanje da li je E zadovoljiv. Ili dokazati da je problem NP-kompletan.

Rešenje:

Ukupan broj literala u `E` je najviše `2n` (za promenljive $x_1, \ldots x_n$)

Tada je `E` oblika:

1. $(x_n + A)(\neg x_n + B)E'$
2. $(x_n + A)E'$
3. $(\neg x_n + A)E'$

Izrazi 2 i 3 su zadovoljivi akko je zadovoljivo `E'`. Izraz 1 je zadovoljiv akko je zadovoljiv `CE'` , gde je `C = A + B` uz izbacivanje ponovljenih promenljivih, ili 1 ako se u `A + B` javlja i promenljiva i njena negacija.
U sva tri slučaja dobijeni izraz zavisi samo od promenljivih $x_1, x_2, \ldots, x_{n-1}$ i svaki literal se javlja najviše jednom. Znači ovim postupkom smo eliminisali jednu promenljivu za polinomijalno vreme O(n), te je odgovarajući algoritam polinomijalne složenosti.