Algebra a diskrétna matematika
Prehľad zo 7. týždňa
Algoritmická zložitosť
Kombinatorika

Algoritmická zložitosť – neformálny výklad

Príklady algoritmických riešení problémov: Zostrojiť kostru daného grafu, alebo rozhodnúť, či daný graf je súvislý, alebo či je hamiltonovský.

V prvom prípade žiadame, aby výstupom algoritmu bol *objekt* – tu graf (kostra), zatiaľ čo v ďalších dvoch prípadoch výstupom algoritmu je len jedna z 2 možných odpovedí: ÁNO – NIE. Takýmto problémom hovoríme *rozhodovacie*.

Uvedomte si, že graf, v ktorom potrebujeme niečo nájsť alebo o ňom urobiť nejaké rozhodnutie, môže mať tisíce vrcholov, je na vstupe algoritmu v podobe napr. zoznamu vrcholov a ich susedov, a najmä to, že počítač ho nevidí!

Odteraz budeme uvažovať len rozhodovacie problémy. Opíšeme základný pojem zložitosti rozhodovacieho problému; na podanie presnej definície zatiaľ nemáme vybudovaný matematický aparát a dozviete sa ju neskôr v špeciálnych predmetoch.

Neformálne, pod **zložitosťou** algoritmického problému budeme rozumieť počet "krokov" $f(n)$, ktoré algoritmus v najhoršom prípade vykoná, aby na výstupe dal odpoveď ÁNO alebo NIE, ak "dĺžka vstupu je n".

"Krok"? Napríklad, v opise algoritmu na prehľadávanie grafu to môže byť inštrukcia 'vezmi ďalší vrchol (hranu) zo zoznamu". V podrobnejšom opise (pseudokóde) to môže byť séria pokynov typu:

(1) Pozri, či vrchol číslo i už bol navštívený.

(2) Ak áno, zvýš honotu i na $i+1$ a vráť sa na (1).

(3) Ak nie, pridaj i do zoznamu prejdených vrcholov a choď na (5).

Atď...

"Krok" v opise algoritmu: $\leq$ *konštantne veľa* "krokov" v pseudokóde.
"Krok" v pseudokóde: $\leq$ *konštantne veľa* "krokov" v program. jazyku.
"Krok" v program. jazyku: $\leq$ *konštantne veľa* inštrukcií v stroj. kóde.
Inštrukcia v strojovom kóde: $\leq$ *konštantne veľa* taktov procesora.

Ak máme prehľadať napr. n-vrcholovú cestu a prejsť každý vrchol, tak:

v opise algoritmu musíme urobiť n "krokov";
v preudokóde to bude viac, ale nie viac ako napr. $5 \cdot n$ krokov,
v strojovom kóde nie viac ako (povedzme) $4 \cdot 5 \cdot n$ krokov,
a celkove (povedzme) nie viac ako $5 \cdot 4 \cdot 5 \cdot n$ taktov.

Či už počet krokov meriame pomocou opisu algoritmu alebo pomocou taktov procesora, dostávame síce rôzne čísla – n a $100n$ – ale stále je to konštanta krát n, kde konštanta závisí od mašiny a implementácie.

V informatike túto situáciu zapíšeme symbolom $O(n)$. Ak teda $f(n)$ z opisu zložitosti je "počet krokov nutných na prejdenie všetkých n vrcholov, tak by sme napísali $f(n) = O(n)$, čo formálne znamená, že $f(n) \leq cn$ pre nejakú konštantu c a pre všetky n. (V tej konštante je zahrnutá naša diskusia o počte krokov.)

Podobne budeme hľadieť na pojem "dĺžka vstupu". Či už je to zoznam vrcholov, ako ho máme na papieri, alebo prekódovaný na postupnosť núl a jednotiek, vždy v prípade napr. cesty na n vrcholoch môžme rovnako dobre povedať, že vstup má dĺžku $O(n)$.

Obvykle sa symbol O používa len na zložitosť, a nie na dĺžku vstupu (to je potom zahrnuté v "narábaní s multiplikatívnymi konštantami").

Matematika má na presné definície týchto pojmov prostriedky – tzv. Turingov stroj; opis jeho činnosti sa dozviete neskôr na špecializovaných prednáškach. Zatiaľ vystačíme s naznačenými intuitívnymi pojmami.

<u>Príklad:</u> Rozhodovací problém zistiť či graf s n vrcholmi a m hranami, zadaný na vstupe, je súvislý. Vzhľadom na našu diskusiu môžme predpokladať, že dĺžka vstupu je $m + n$. Počet krokov prehľadávacieho algoritmu je $O(m + n)$, pretože každú hranu navštívime najviac raz (niektoré vrcholy viackrát, ale to je už započítané v člene m).

Niekedy sa zložitosť udáva *len* v terminológii počtu vrcholov n grafu. Keďže $m \leq n(n-1)/2$, môžme jednoducho povedať, že zložitosť prehľadávacieho algoritmu je nanajvýš $O(n^2)$. Podstatné je, že n^2 je *polynóm* v premennej n. Je to rastúca funkcia, ale nie "divoko" rastúca.

Algoritmy, ktoré na rozhodovacie problémy o grafoch s n vrcholmi dajú odpoveď ÁNO alebo NIE a spotrebujú pri tom najviac $O(n^k)$ krokov pre *konštantné* k, sa nazývajú **polynomiálne**, alebo **patriace do triedy** P.

Príklady: Určenie, či priemer grafu je $\leq d$ pre dané d, rozhodnutie, či graf má perfektné párovanie, alebo či daný graf je rovinný, atď.

Je však celý rad rozhodovacích problémov, na ktoré zatiaľ nepoznáme žiaden polynomiálny algoritmus.

Príklady: Rozhodnúť, či daný graf s n vrcholmi na vstupe je hamiltonovský, alebo či jeho chromatické číslo je $\leq \ell$ pre ľubovoľné *konštantné* $\ell \geq 3$.

Aj tie najlepšie známe algoritmy na tieto úlohy dokážu spracovať vstupný graf s n vrcholmi v najlepšom prípade až po $O(c^n)$ krokoch, kde $c > 1$
V informatike sa skúma celá trieda takýchto problémov, ktoré možno "rýchlo" – v polynomiálnom čase – pretransformovať jeden na druhý, a v konečnom dôsledku na rozhodnutie, či daný n-vrcholový graf na vstupe je hamiltonovský.

Tejto triede problémov sa hovorí **NP-úplné problémy**. Ich definícia vysoko presahuje túto prednášku (NP znamená *nedeterministické polynomiálne*).

Rozdiel medzi zložitosťou napr. $O(n^3)$ a $O(2^n)$: Ak napr. $n = 400$, tak (až na multiplikatívnu konštantu) porovnávame 400^3 s 2^{400}; to druhé je viac, ako fyzikmi odhadovaný počet elementárnych častíc vo vesmíre!

P-NP problém: Jeden z najslávnejších problémov v súčasnosti zo zoznamu 7 miléniových problémov Clayovho inštitútu Princetonskej univerzity. Vyriešenie každého z nich je dotovaný miliónom USD!
(Jeden z nich už je vyriešený.)

P-NP problém je otázka, či P=NP;
v ekvivalentnej formulácii, či existuje algoritmus polynomiálnej zložitosti na rozhodnutie, či ľubovoľný daný n-vrcholový graf je hamiltonovský. (Vzhľadom na ekvivalentnosť v triede NP-úplných problémov by kladná odpoveď znamenala existenciu polynomiálnych algoritmov pre *všetky* NP-úplné problémy.)

Verí sa, že P$\neq$NP, ale nikto to nevie dokázať! Tu je významný problém *izomorfizmu* grafov: Rozhodnúť, či dva n-vrcholové grafy na vstupe sú izomorfné. Zatiaľ nepoznáme žiaden polynomiálny algoritmus na tento problém, ale na druhej strane ani nikto nevie dokázať, že je NP-úplný!

Predpokladá sa, že ide o problém, ktorý striktne medzi P a NP! Ak by to niekto dokázal, tak by zároveň vyriešil aj P-NP problém.

Kombinatorika

Kombinatorika sa zaoberá konečnými množinami, ich štruktúrami, usporiadaním, rozkladom na menšie objekty, zobrazeniami medzi nimi, usporiadanými n-ticami, atď.

Tvrdenie 1: Ľubovoľná n-prvková množina má práve 2^n podmnožín.

Dôkaz: Nech $A = \{a_1, a_2, a_3, \dots, a_n\}$.

Každú podmnožinu B množniny A môžme reprezentovať pomocou n-tice 0 a 1, pričom

na i-tej pozícii je $\begin{cases} 1 & \text{ak} \quad a_i \in B \\ 0 & \text{ak} \quad a_i \notin B \end{cases}$

Napr. $B = \{a_1, a_3, a_4\} = (1, 0, 1, 1, \dots)$

Každá podmnožina množiny A má jednoznačnú reprezentáciu pomocou n-tice núl a jednotiek. Celkový počet rôznych n-tíc núl a jednotiek je 2^n, čo je aj hľadaný počet všetkých podmnožín množiny veľkosti n. $\square$

Tvrdenie 2: Každá n-prvková množina má práve 2^{n-1} podmnožín nepárnej veľkosti a 2^{n-1} podmnožín párnej veľkosti.

Dôkaz: Nech A je n-prvková množina a prvok $a \in A$.

Z predchádzajúceho tvrdenia vieme, že počet všetkých podmnožín množiny $A - \{a\}$ je 2^{n-1}.

Vyberme si ľubovoľnú z nich, $B \subseteq A - \{a\}$.

Ak B má nepárny počet prvkov, je to aj želaná podmnožina množiny A s nepárnym počtom prvkov.

Ak B má párny počet prvkov, pridáme k nej prvok a.
Potom $B \cup \{a\} \subseteq A$ a veľkosť $|B \cup \{a\}|$ je nepárna.

Našli sme bijekciu medzi množinou všetkých podmnožín $A-\{a\}$ a množinou všetkých podmnožín A nepárnej veľkosti. Je ich 2^{n-1}.

Doplnok k nim sú všetky podmnožiny párnej veľkosti: $2^n - 2^{n-1} = 2^{n-1}$. $\square$

Variácie k-tej triedy z n prvkov s opakovaním

- všetky možné usporiadané výbery k prvkov z n prvkov, pričom vo výberoch sa prvky *môžu opakovať*
- všetky zobrazenia z k-prvkovej množiny do n-prvkovej množiny
- počet slov dĺžky k nad abecedou z n písmen

Ich počet je

$$\mathbf{V^*(n,k) = n^k}$$

Na každú "pozíciu" 1, 2, ..., k možno vybrať ktorýkoľvek z n prvkov.

Príklad 1: Koľko rôznych PIN-kódov si môžte zvoliť pre bankovú kartu?

$$V^*(10,4) = 10000$$

Príklad 2: Koľko rôznych kódov dĺžky 5 môžte vytvoriť z písmen A, E, I, O, U, Y?

$$V^*(6,5) = 6^5 = 7776$$

Príklad 3: Koľko existuje rôznych ŠPZ vozidiel ku každému označeniu mesta?

$$V^*(10,3) \cdot V^*(24,2) = 10^3 \cdot 24^2 = 576000$$

Príklad 4: Koľko párnych 5-ciferných čísel môžme napísať z cifier $0,1,2,3,4,5,6$?

$$6 \cdot 7 \cdot 7 \cdot 4 = 1176$$

Variácie k-tej triedy z n prvkov bez opakovania

• všetky možné usporiadané výbery *navzájom rôznych* k prvkov z n prvkov

• všetky *prosté* zobrazenia z k-prvkovej množiny do n-prvkovej množiny

• počet slov dĺžky k z navzájom rôznych písmen nad abecedou z n písmen

Ich počet je

$$\mathbf{V(n,k) = n(n-1)...(n-(k-1)) = \frac{n!}{(n-k)!}}$$

Na "pozície" 1, 2, ..., k možno postupne vybrať ktorýkoľvek z n prvkov na pozíciu 1, ktorýkoľvek zo zvyšných $n-1$ prvkov na pozíciu 2, atď, až napokon (keď už aj $(k-1)$ vá pozícia je obsadená) ktorýkoľvek zo zvyšných $(n-(k-1))$ prvkov na pozíciu k.

<u>Príklad 5</u>: Koľko rôznych 5- písmenových slov sa dá zostaviť z písmen slova VYHRAŤ, ak sa žiadne neopakuje?

$$V(6,4) = 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 = 720$$

<u>Príklad 6</u>: Koľko rôznych umiestnení na prvých troch miestach je možných v súťaži s 10 účastníkmi?

$$V(10,3) = 10 \cdot 9 \cdot 8 = 720$$

<u>Príklad 7</u>: Koľko je rôznych 4-ciferných párnych čísel, v ktorých sú všetky cifry rôzne?

$$8 \cdot 8 \cdot 7 \cdot 4 + 9 \cdot 8 \cdot 7 \cdot 1 = 2296$$

Permutácia n prvkov

• *variácia* n-tej triedy z n prvkov bez opakovania

• ľubovoľná *bijekcia* n-prvkovej množiny

• počet slov dĺžky n z navzájom rôznych písmen nad abecedou z n písmen

Ich počet je

$$\mathbf{P(n) = V(n,n) = n! = \prod_{i=1}^{n} i}$$

Príklad 8: Koľko rôznych slov dĺžky 6 je možné vytvoriť z písmen slova PIATOK?

$$6! = 720$$

Príklad 9: Koľkými rôznymi spôsobmi je možné usadiť do radu 10 ľudí?

$$10! = 3628800$$

Príklad 10: Aký je počet variácií k-tej triedy z množiny $\{1, 2, \ldots, n\}$ bez opakovania a permutácii z množiny $\{1, 2, \ldots, n\}$ takých, že 1 a 2 nie sú vedľa seba?

$$V(n,k) - 2(k-1)V(n-2,k-2);\ \text{ pre permutácie } k = n$$

Príklad 11: Aký je počet podmnožín množiny $\{1, 2, \ldots, n\}$, ktoré obsahujú všetky nepárne čísla $\leq n$?

$$2^{\lfloor \frac{n}{2} \rfloor}$$

Príklad 12: Koľkými spôsobmi je možné rozdeliť množinu $\{1, 2, \ldots, n\}$ na 2 disjunktné podmnožiny, ak nezáleží na poradí podmnožín?

$$2^{n-1}$$

Príklad 13: Pre n=5 jedna možná premutácia je

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 5 & 2 & 4 \end{pmatrix}$$

$$p(1) = 3,\ p(2) = 1,\ p(3) = 5,\ p(4) = 2,\ p(5) = 4$$

Kratší zápis pomocou *cyklu*: $p = (13542)$

Príklad 14: Pomocou cyklov zapíšte permutáciu

$$p = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 7 & 1 & 6 & 8 & 9 & 4 & 2 & 3 & 5 \end{pmatrix} = (172)(3648)(59)$$

Identická permutácia – $id = (1)(2)(3) \cdot (n)$.

Skladanie permutácií vykonávame *zľava doprava*.

Príklad 15: Zložte dané permutácie

$(12)(34) \circ (13)(24) = (14)(23)$

$(13)(24) \circ (12)(34) = (14)(23)$

$(134)(258) \circ (2456)(78) = (135784)(62)$

$(2456)(78) \circ (134)(258) = (134872)(56)$

Vo všeobecnosti je skladanie permutácií nekomutatívne, ale máme výnimky.

Kombinácie k-tej triedy z n prvkov

- všetky možné *neusporiadané* výbery *navzájom rôznych* k prvkov z n prvkov
- všetky možné k-prvkové *podmnožiny* n-prvkovej množiny

Ich počet dostaneme z variácií k-tej triedy bez opakovania vydelením $k!$, čo je počet všetkých usporiadaní konkrétnej variácie, t.j.

počet kombinácií k-tej triedy z n prvkov je

$$\mathbf{C(n,k) = \frac{n(n-1)...(n-(k-1))}{k!} = \frac{n!}{(k!(n-k)!)} = \binom{n}{k}}$$

Vlastnosť 1:

$$\binom{n}{k} = \binom{n}{n-k}$$

Vlastnosť 2:

$$\binom{n-1}{k-1} + \binom{n-1}{k} = \binom{n}{k}$$

Dôkaz: Pravá strana je počet k-prvkových podmnožín n-prvkovej množiny A. Zvoľme si $a \in A$. Podmnožiny množiny A si rozdeľme podľa toho, či obsahujú a alebo nie.

Každá k-prvková podmnožina množiny A neobsahujúca a je zároveň aj k-prvková podmnožina množiny $A - \{a\}$. Všetkých takých podmnožín je $\binom{n-1}{k}$.

Ak B je nejaká k-prvková podmnožina A obsahujúca a, môžeme jej bijektívne priradiť $(k-1)$-prvkovú podmnožinu množiny $B - \{a\}$.

Ich počet je $\binom{n-1}{k-1}$. Sčítaním týchto dvoch kombinačných čísel dostaneme dokazovanú rovnosť.

$\square$

Pascalov trojuholník

$$
\begin{array}{ccccccccccccccccccccc}
&&&&&&&&&&1&&&&&&&&&& \\
&&&&&&&&&1&&1&&&&&&&&& \\
&&&&&&&&1&&2&&1&&&&&&&& \\
&&&&&&&1&&3&&3&&1&&&&&&& \\
&&&&&&1&&4&&6&&4&&1&&&&&& \\
&&&&&1&&5&&10&&10&&5&&1&&&&& \\
&&&&1&&6&&15&&20&&15&&6&&1&&&& \\
&&&1&&7&&21&&35&&35&&21&&7&&1&&& \\
&&1&&8&&28&&56&&70&&56&&28&&8&&1&& \\
&1&&9&&36&&84&&126&&126&&84&&36&&9&&1& \\
1&&10&&45&&120&&210&&252&&210&&120&&45&&10&&1 \\
&\iddots&&&&&&&&&\vdots&&&&&&&&&\ddots&
\end{array}
$$

Vlastnosť 3 (Binomická veta):

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

Vlastnosť 4:

$$(x+1)^n = \sum_{k=0}^{n} \binom{n}{k} x^k$$

<u>Dôkaz:</u> Matematickou indukciou vzhľadmon na n.

1. Vzťah platí pre $n = 0$.

2. Predoklajme, že tvrdenie je splnená pre nejaké $n \geq 0$. Našou úlohou teraz je, dokázať, že rovnica platí aj pre $n+1$.

$$(x+1)^{(n+1)} = (x+1)(x+1)^n = (1+x)\sum_{k=0}^{n}\binom{n}{k}x^k = \sum_{k=0}^{n}\binom{n}{k}x^k + \sum_{k=0}^{n}\binom{n}{k}x^{(k+1)} =$$

$$= \binom{n}{0}x^0 + \sum_{k=1}^{n}\binom{n}{k}x^k + \sum_{k=0}^{n-1}\binom{n}{k}x^{(k+1)} + \binom{n}{n}x^{n+1} =$$

$$= 1 + \sum_{k=1}^{n}\binom{n}{k}x^k + \sum_{k=1}^{n}\binom{n}{k-1}x^k + x^{n+1} =$$

$$= 1 + \sum_{k=1}^{n}\left(\binom{n}{k} + \binom{n}{k-1}\right)x^k + x^{n+1} =$$

$$= \binom{n+1}{0}x^0 + \sum_{k=1}^{n}\binom{n+1}{k}x^k + \binom{n+1}{n+1}x^{n+1} = \sum_{k=0}^{n+1}\binom{n+1}{k}x^k$$

$\square$

Vlastnosť 5:

$$\sum_{k=0}^{n}\binom{n}{k} = \binom{n}{0} + \binom{n}{1} + \ldots + \binom{n}{n-1} + \binom{n}{n} = 2^n$$

Vlastnosť 6:

$$\sum_{j=0}^{r}\binom{m}{j}\binom{n}{r-j} = \binom{m+n}{r}$$

Vlastnosť 7:

$$\sum_{j=0}^{n}\binom{n}{j}^2 = \binom{2n}{n}$$