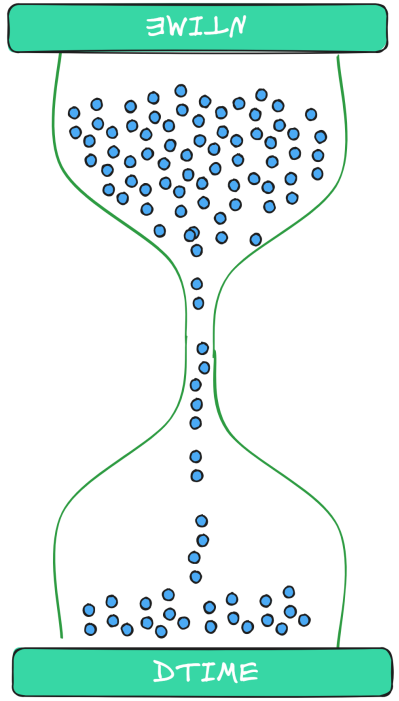
Časovna zahtevnost

Uroš Čibej



Pregled

- Modeliranje časovne zahtevnosti
- Relacije med računskimi modeli
- Razred P
- Razred NP

Literatura

- Sipser razdelek 7
- **dodatno:** Intro TCS - Predavanje 10

Pregled problemov

Problem klike

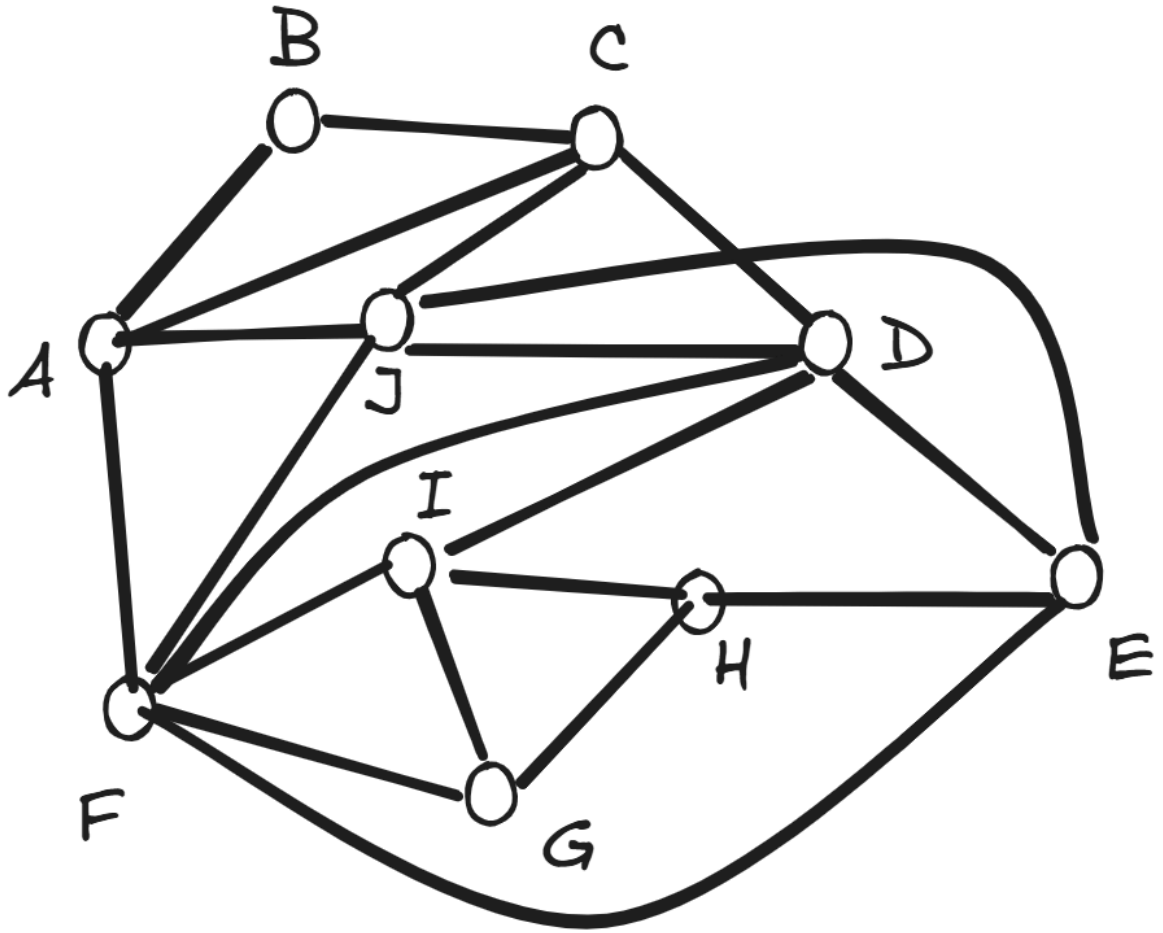
Vhod: $G = \langle E, V \rangle, k$

Vprašanje: Ali obstaja $V' \subseteq V, |V'| \geq k,$

$$\forall u, v \in V' : v \neq u, (u, v) \in E$$

Primer

- Ali obstaja klika velikosti 3?
- Ali obstaja klika velikosti 4?
- Ali obstaja klika velikosti 5?



Problem prereza grafa

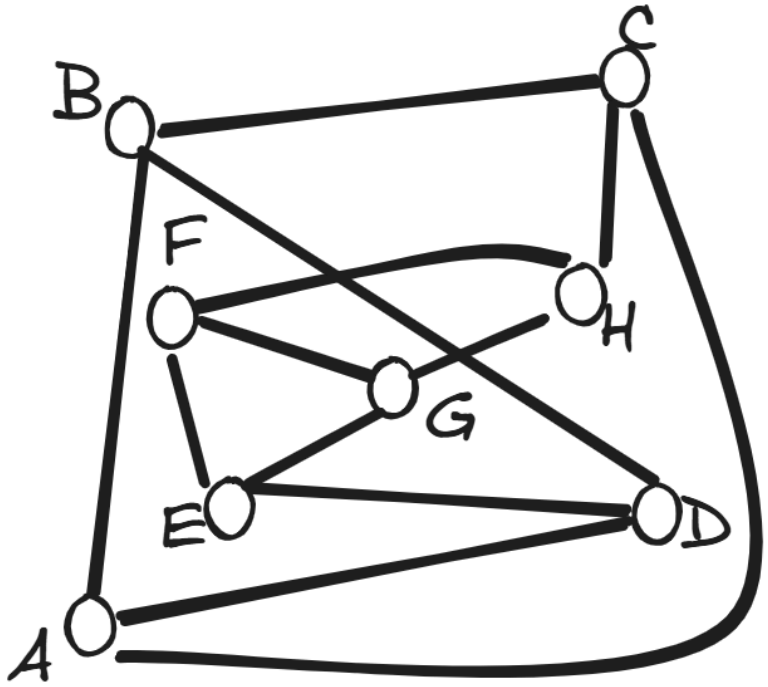
Vhod: $G = \langle E, V \rangle, k$

Vprašanje: Ali obstaja razbitje V na $V', V \setminus V'$, da

$$C = \{(u, v) \in E \mid u \in V', v \in V \setminus V'\}, |C| = k$$

Primer

- Ali obstaja prerez velikosti 4?
- Ali obstaja prerez velikosti 3?
- Ali obstaja prerez velikosti 2?



Vsota podmnožice

Vhod: Končna množica $A \subset \mathrm{N}$, $k \in \mathrm{N}$

Vprašanje: Ali obstaja $A' \subseteq A$, da velja:

$$\sum_{a \in A'} a = k$$

Primer

$$A = \{81, 79, 63, 58, 41, 35, 27, 21, 19, 11\}, k = 199$$

Problem SAT (satisfiability)

Vhod: Množica logičnih spremenljivk $X = \{x_1, x_2, \ldots, x_n\}$ in logični izraz s temi spremenljivkami:

$$\Phi$$

Vprašanje: Ali obstaja dodelitev vrednosti spremenljivkam $\tau : X \to \{0, 1\}$, da je

$$\Phi[\tau] = 1$$

Primer

$$
\begin{aligned}
\Phi = \; & C_1 : && (x_1 \vee x_2 \vee x_3) \\
\wedge & C_2 : && (\neg x_1 \vee x_4) \\
\wedge & C_3 : && (\neg x_2 \vee x_4) \\
\wedge & C_4 : && (\neg x_3 \vee x_5) \\
\wedge & C_5 : && (\neg x_4 \vee \neg x_5) \\
\wedge & C_6 : && (x_2 \vee \neg x_3 \vee \neg x_4) \\
\wedge & C_7^* : && (\neg x_4)
\end{aligned}
$$

Vprašanja

- Kako oceniti porabo virov za reševanje problemov?
- Kateri problemi so hitreje/počasneje rešljivi?
- Kako so ti problemi med seboj podobni/različni?

Časovna zahtevnost TS

def. Časovna zahtevnost deterministočnega stroja M je funkcija $f : \mathrm{N} \to \mathrm{N}$, kjer je $f(n)$ največje število korakov, ki ga naredi M pri nekem vhodu dolžine n.

Primer

$$L = \{0^i 1^i\}$$

Veliki O

def. Naj bosta f in g funkciji. $f(n) = O(g(n))$ če obstajata konstanti c, n_0, da za vsak $n \geq n_0$

$$f(n) \leq cg(n)$$

Razred časovne zahtevnosti

def. Razred računske zahtevnosti $TIME(t(n))$ je množica jezikov za katere obstaja Turingov stroj s časovno zahtevnostjo $O(t(n))$

Primer

$$L = \{0^i 1^i\}$$

1. $L \in TIME(n^2)$
2. $L \in TIME(n \log n)$
3. $L \in TIME(n)$

Različni modeli

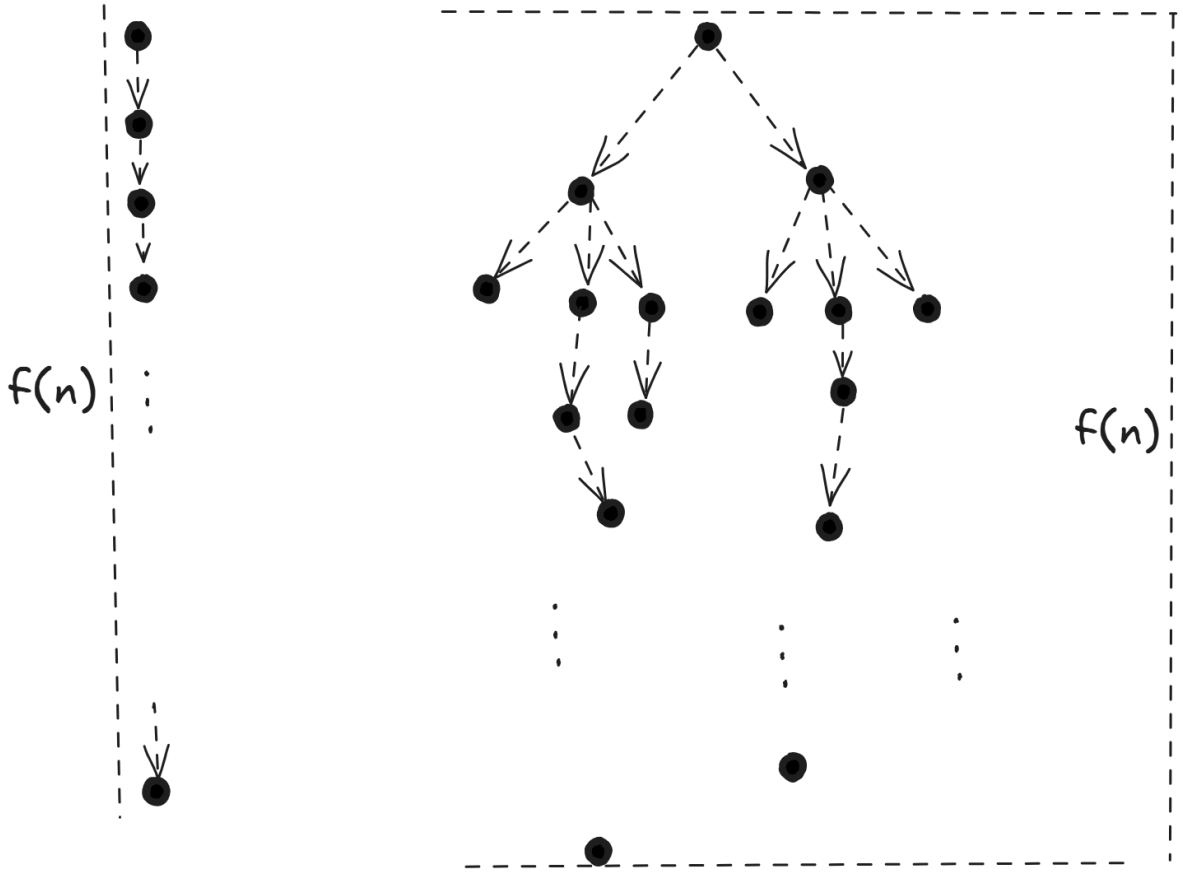
- večtračni : enotračni
- nedeterministični : deterministični

Simulacija več trakov

Izrek: Vsak večtračni TS M s časovno zahtevnostjo $t(n)$ ima ekvivalenten enotračni TS s časovno zahtevnostjo $t^2(n)$

Časovna zahtevnost nedeterminističnega TS

def. Časovna zahtevnost nedeterminističnega stroja M je funkcija $f : \mathrm{N} \to \mathrm{N}$, kjer je $f(n)$ največje število korakov, ki ga naredi M pri poljubni veji drevesa izvajanja pri nekem vhodu dolžine n.



Razred nedeterministične časovne zahtevnosti

def. Razred računske zahtevnosti $NTIME(t(n))$ je množica jezikov za katere obstaja nedeterministični Turingov stroj s časovno zahtevnostjo $O(t(n))$

Simulacija nedeterminizma

Izrek: Vsak **nedeterministični enotračni** TS M s časovno zahtevnostjo $t(n)$ ima ekvivalenten **enotračni deterministični** TS s časovno zahtevnostjo $2^{O(t(n))}$

Razred P - definicija

$$P = \bigcup_k TIME(n^k)$$

Razred P (zakaj polinomsko = dobro)

1. P je invarianten za različne modele deterministčnih strojev
2. P v grobem ustreza razredu problemov, ki so v praksi obvladljivi

Primeri problemov v P

- preverjanje praštevilskosti
- vsi kontekstno neodvisni jeziki
- poti v grafu
- prerez grafa

Nedeterminizem in časovna zahtevnost

Preverjevalnik

def. Preverjevalnik V je algoritem za jezik

$$A = \{w \mid V \text{ sprejme } \langle w, c \rangle \text{ za nek niz } c\}$$

Primer

Sestavljena števila

$$L = \{n \mid n \text{ je sestavljeno število}\}$$

Primer

Hamiltonov cikel

$$H = \{\langle G \rangle \mid G \text{ ima Hamiltonov cikel}\}$$

Definicija razreda NP

def. NP je razred jezikov, ki imajo **polinomski preverjevalnik**.

NP in nedeterminizem

Izrek $L \in NP \iff$ ga razpoznava polinomski nedeterministični stroj.

$L \in NP \implies$ polinomski nedeterministični stroj M

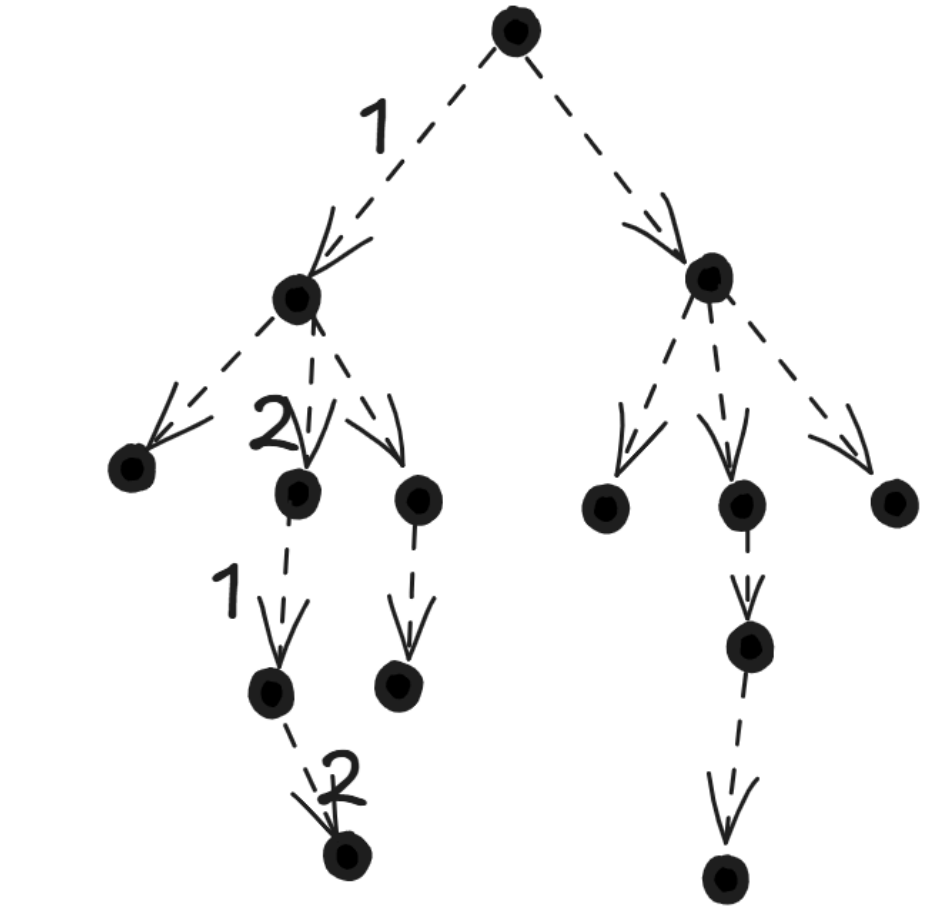
Naj bo V polinomski preverjevalnik za L

M deluje tako:

1. nedeterministično generira c
2. vrne $V(w, c)$

polinomski nedeterministični stroj $\Longrightarrow$ $L \in NP$

$V(w, c)$ skonstruiramo tako, da simulira M, ob vsaki nedeterministični odločitvi pa uporabi en znak iz c.


c=1212....2
1
2
1
2

Alternativna definicija NP

$$NP = \bigcup_k NTIME(n^k)$$

Primeri problemov v NP

In ni znano, če so v P

- Klika
- Vsota podmnožice
- SAT
- ...

Vprašanje P proti NP

- Ali je preverjanje rešitve enako zahtevno kot generiranje rešitve?
- Ali je izčrpno preiskovanje (eksponentno) edini pristop, ki deluje za določene vrste problemov?
- Se moramo pri določenih problemih zadovoljiti z neoptimalnimi rešitvami?