Problém obchodného cestujúceho (TSP)

Úloha (TSP): Daný je ohodnotený neorientovaný graf $G = (V, E)$
Nájdite “okružnú” cestu daným grafom:

- Začne a skončí v tom istom vrchole
- Všetky ostatné vrcholy navštívi práve raz
- Najkratšia možná

Na túto úlohu sú metódy, ktoré sme doteraz spomínali, krátke.

- **Túto úlohu neviem riešiť v polynomiálnom čase.**

  Nie príliš dobrý argument.

- **Pre túto úlohu neexistuje algoritmus, ktorý by ju riešil v polynomiálnom čase.**

  Nie príliš realistické pre väčšinu úloh.

- **Túto úlohu síce neviem riešiť v polynomiálnom čase, no veľa veľmi múdrych ľudí to už skúšalo a tiež nič rozumné nevymyslelo**

  Aká je šanca, že sa veľa múdrych ľudí desiatky rokov zaoberalo priamo vašou úlohou?

  Trieda tzv. **NP-ťažkých** resp. **NP-úplných** problémov.

NP ťažkosť a úplnosť: osnova

- Rozhodovacie vs. optimalizačné problémy.
- Trieda problémov P.
- Nedeterministické výpočty a trieda problémov NP.
- Cookova veta: Existuje NP-úplný problém.
- Ako ukázať, že problém je NP-úplný?
- Základné "portfólio" NP-úplných problémov.

Rozhodovacie problémy

Rozhodovací problém: Taký čo má odpoveď áno / nie

Optimalizačný problém:
TSP: Nájdi v grafe okružnú cestu s minimálnou dĺžkou

Zodpovedajúci rozhodovací problém:
TSP-D: Existuje v grafe okružná cesta s dĺžkou nanajvýš B?

Pomocou rozhodovacieho problému vieme vyriešiť aj optimalizačný:

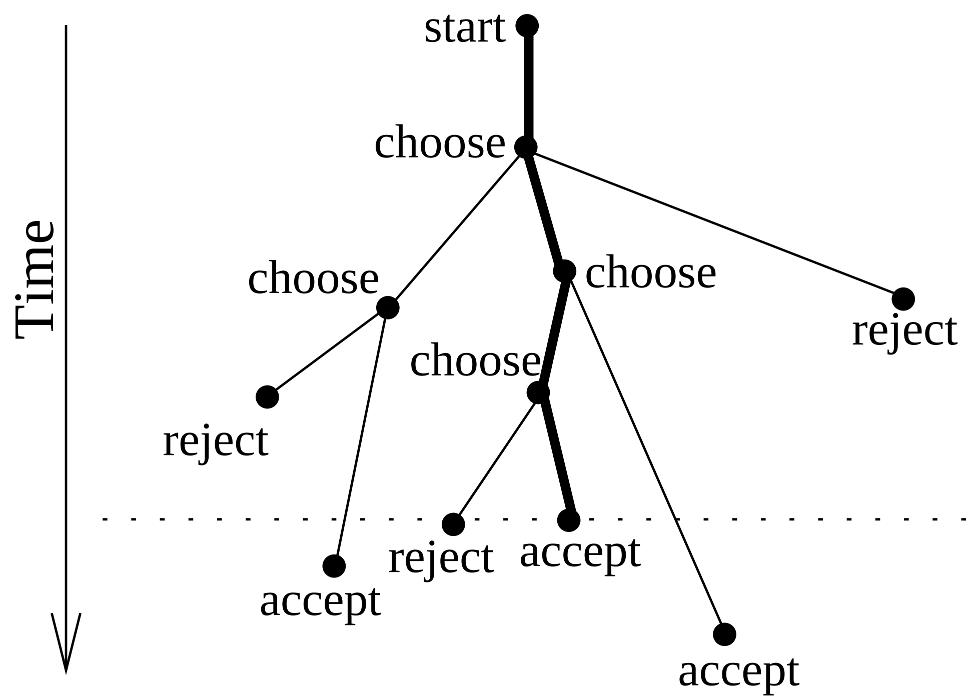
- Minimálnu dĺžku viem nájsť napríklad binárnym vyhľadávaním
- Konkrétnu cestu viem zrekonštruovať pomocou dotazov na mierne modifikované grafy.

(Viac na cvičeniach.)

Nedeterministické výpočty

Nové konštrukty do našich pseudokódov:

- **accept:** skonči výpočet a odpovedz "áno"
- **reject:** skonči výpočet a odpovedz "nie"
- **choose** k **from** $[i, j]$**:** nastav hodnotu k na hodnotu medzi i a j tak, že sa dostaneme k "áno" najkratším možným výpočtom (ak všetky výpočty vedú k odpovedi "nie", nastav k na ľubovoľnú hodnotu z $[i, j]$)



Poznámka: Nedeterministické algoritmy vieme naimplementovať na deterministickom stroji za cenu exponenciálnej časovej zložitosti (backtracking cez strom)

Nedeterministický algoritmus pre riešenie TSP-D

```
function TSP-D
  visited[i]:=false for all vertices;
  last_visited:=1; visited[1]:=true;
  length:=0;
  repeat n-1 times
    choose next_visited from [1,n] ;
    if visited[next_visited] then reject;
    //we cannot visit a single vertex twice
    visited[next_visited]:=true;
    length:=length+w(last_visited,next_visited);
    last_visited:=next_visited;
  length:=length+w(last_visited,1);
  if length<=B then accept;
                else reject;
```

Triedy P a NP

- **Trieda P:** Rozhodovacie problémy, pre ktorých riešenie existuje algoritmus, ktorý beží v **deterministickom polynomiálnom čase**.
- **Trieda NP:** Rozhodovacie problémy, pre ktorých riešenie existuje algoritmus, ktorý beží v **nedeterministickom polynomiálnom čase**.
- Zjavne $P \subseteq NP$.
- Existujú problémy, ktoré sú v triede NP a nie sú v triede P?

  $1M otázka (slávny P vs. NP problém)

  Všeobecne sa verí, že $\mathrm{P} \neq \mathrm{NP}$
- **NP-ťažký problém:** Ak by sme vedeli vyriešiť hociktorý NP-ťažkýc problém v deterministickom polynomiálnom čase, vedeli by sme vyriešiť **všetky** problémy z NP v polynomiálnom čase.
  Ak problém navyše patrí do NP, tak ho voláme **NP-úplný.**
  TSP je NP-ťažký. TSP-D je NP-úplný.

Hamiltonovská kružnica (HAM)

Rozhodovací problém: Daný je neohodnotený neorientovaný graf
Existuje okružná cesta, ktorá prechádza všetkými vrcholmi práve raz?

HAM patrí do NP:

```
visited[i]:=false for all vertices;
last_visited:=1; visited[1]:=true;
length:=0;
repeat n-1 times
  choose next_visited from [1,n];
  if visited[next_visited] then reject;
  if not edge(last_visited,next_visited) then reject;
  //we cannot visit a single vertex twice
  visited[next_visited]:=true;
  last_visited:=next_visited;
if edge(last_visited,1) then accept;
else reject;
```

Splniteľnosť (SAT) a 3-splniteľnosť (3-SAT)

Rozhodovací problém: Daná je logická formula f nad n premennými. Existuje priradenie pravdivostných hodnôt také, že formula f je splnená?

V prípade 3-SAT: Formula je tvaru:
$(a_{1,1} \vee a_{1,2} \vee a_{1,3}) \wedge (a_{2,1} \vee a_{2,2} \vee a_{2,3}) \wedge \cdots \wedge (a_{m,1} \vee a_{m,2} \vee a_{m,3})$
$a_{i,j}$ je x_k alebo $\neg x_k$

SAT, 3-SAT patrí do NP:

```
for i:=1 to n
  choose x[i] from [0,1]
if assignment x satisfies formula f
  return accept
else
  return reject
```

Najmenšie vrcholové pokrytie (VC)

Optimalizačný problém VC: Daný je neorientovaný neohodnotený graf $G = (V, E)$. Nájdite najmenšiu množinu vrcholov $C \subseteq V$ takú, že každá hrana z E je "pokrytá" (aspoň jeden jej vrchol je v C).

Rozhodovací problém VC-D: Existuje vrcholové pokrytie C také, že jeho veľkosť $|C| \leq c$?

VC-D patrí do NP:

```
for i:=1 to n
  choose x from [0,1]
  if x then C:=C+{i}
if (size(C)>c) reject;
if (C is not a vertex cover) reject;
else accept;
```

Najväčšia klika v grafe (CLIQUE)

Optimalizačný problém CLIQUE: Daný je neorientovaný neohodnotený graf $G = (V, E)$. Nájdite najväčšiu množinu vrcholov $C \subseteq V$ takú, že vrcholy z C tvoria v grafe kliku.

Rozhodovací problém CLIQUE-D: Existuje v grafe klika o veľkosti aspoň k?

CLIQUE-D patrí do NP:

```
for i:=1 to n
  choose x from [0,1]
  if x then C:=C+{i}
if (size(C)<k) reject;
if (C is not a clique) reject;
else accept;
```

Zhrnutie

- Problém obchodného cestujúceho (TSP) a jeho rozhodovacia verzia (TSP-D)
- Idea nedeterministických výpočtov, nedeterministická polynomiálna časová zložitosť
- Zložitostné triedy P a NP, otvorený problém $\mathrm{P} \stackrel{?}{=} \mathrm{NP}$
- (Neformálna) definícia NP-ťažkých a NP-úplných problémov
- (Bez dôkazu) Paleta NP-úplných problémov: TSP-D, HAM, VC-D, SAT, 3-SAT, CLIQUE-D
- Na budúce: Dokazovanie, že problém je NP-úplný
- Ak o probléme dokážeme, že je NP-úplný, potom je veľmi ťažké očakávať, že ho budeme vedieť riešiť v deterministickom polynomiálnom čase.