Juraj Holas

# Domáca úloha č. 4

## 1. Polynomiálne redukcie

#### a) HAM → TSP-D

Vstupný graf pre HAM sa na vstupný graf pre TSP-D zmení nasledovne:

4

3

2

1

1

1

0

0

0

0

4

3

2

1

Vstup pre TSP-D je teda graf a maximálna suma rovná 0.

Algoritmus pre TSP-D vráti odpoveď „áno“, pričom príslušná obchôdzka obchodného cestujúceho je postupnosť vrcholov v . Príslušná Hamiltonovská kružnica v  je tiež postupnosť vrcholov .

#### b) 3-SAT → VC

Vstupná formula sa na vstupný graf pre VC zmení nasledovne (zatiaľ nie je rozdiel medzi farebne odlíšenými vrcholmi) :

Zadaná formula je splniteľná práve vtedy, keď graf má VC veľkosti nanajvýš , kde je počet logických premenných, a počet klauzúl. V našom prípade teda potrebujeme nájsť VC, ktorý použije najviac 12 vrcholov. Takýto VC existuje, a príslušných 12 vrcholov je na obrázku vyznačených červenou farbou.

Ohodnotenie logických premenných, ktoré korešponduje nájdenému VC sú teda tie vrcholy, ktoré boli vybrané, v našom prípade je formula splnená ak platí: .

#### c) 3-SAT → SUBSET-SUM

Vstupná formula sa zmení na nasledovné čísla v množine :

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  | **1** | 0 | 0 | 0 | 0 | **1** | 0 | 0 |
|  | **1** | 0 | 0 | 0 | 0 | 0 | **1** | 0 |
|  | 0 | **1** | 0 | 0 | 0 | 0 | **1** | 0 |
|  | 0 | **1** | 0 | 0 | **1** | **1** | 0 | **1** |
|  | 0 | 0 | **1** | 0 | **1** | **1** | 0 | 0 |
|  | 0 | 0 | **1** | 0 | 0 | 0 | 0 | **1** |
|  | 0 | 0 | 0 | **1** | 0 | 0 | **1** | **1** |
|  | 0 | 0 | 0 | **1** | **1** | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | **1** | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | **2** | 0 | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | **1** | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | **2** | 0 | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | **1** | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | **2** | 0 |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **2** |

Cieľová suma .

Pre takúto množinu vieme nájsť podmnožinu, ktorej súčet bude rovný , napr. (zvýraznené riadky). Zodpovedajúce ohodnotenie logických premenných, pri ktorom bude formula splnená, je určené prvkami vo vybranej podmnožine. Pre splnenie formuly teda musí platiť .

#### d) SUBSET-SUM → COIN

Vstupná množina sa zmení na nasledovné hodnoty mincí:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | **1** | **1** | 0 | 0 | 0 |
|  | 0 | **1** | 0 | 0 | 0 |
|  | **4** | 0 | **1** | 0 | 0 |
|  | 0 | 0 | **1** | 0 | 0 |
|  | **5** | 0 | 0 | **1** | 0 |
|  | 0 | 0 | 0 | **1** | 0 |
|  | **6** | 0 | 0 | 0 | **1** |
|  | 0 | 0 | 0 | 0 | **1** |

Pôvodná cieľová hodnota sa zmení na cieľovú sumu . Túto sumu však z uvedených mincí nevieme vyskladať, a teda ani v pôvodnom probléme SUBSET-SUM neexistuje taká podmnožina , aby jej súčet bol rovný .

Predpokladajme, že sumu vieme z mincí vyskladať. To znamená, že vieme nájsť koeficienty také, aby:

Prvá cifra výslednej sumy sa rovná (samozrejme, pri číselnej sústave s vhodným základom). Na prvej cifre sumy (ak chceme dostať iba 5-ciferné číslo) avšak nevieme dostať cifru inak, než že aspoň jedno z  bude väčšie ako . Tým by sa nám ale zvýšila hodnota aj v nejakej z ďalších cifier na alebo viac, čiže výsledné číslo by sa nerovnalo . Tým sme dospeli k sporu. To znamená, že nevieme nájsť také koeficienty , aby prvá cifra výsledku bola rovná , a teda nevieme nájsť takú podmnožinu , aby súčet jej členov bol rovný .

## 2. Znova animácie

Pózy si môžeme predstaviť ako vrcholy a animácie ako orientované, váhované hrany grafu, kde hrana predstavuje vykonanie animácie v originálnom smere, a hrana v vykonanie animácie odzadu. Začiatočná a konečná póza každej animácie je teda spojená dvojicou hrán – jednou v každom smere. V takto skonštruovanom grafe je teda Jankovou úlohou nájsť ťah dĺžky začínajúci vo vrchole a končiaci v .

#### Riešenie v NP čase:

Problém sa dá riešiť v NP čase nasledujúcim nedeterministickým algoritmom:

1 function animation(graph G, vertex start, vertex finish, number k){

2 G = convert\_to\_neighbour\_table(G)

3 v = start

4 sum = 0

5 used = {};

6 while(true){

7 u = choose from v.neighbours

8 if(edge(v, u) in used){ reject }

9 else{ used.add(edge(v, u)) }

10

11 sum += edge(v, u).weight

12 if(sum > k){ reject }

13 if(u == finish && sum == k){ accept }

14

15 v = u

16 }

17 reject

18 }

Časová zložitosť algoritmu ( je počet vrcholov, počet hrán):

* Konverzia grafu do tabuľky susedov (r. 2) potrebuje vytvoriť tabuľku záznamov o veľkosti bitov, pričom túto tabuľku môže vytvoriť „priamočiaro“ časová zložitosť je teda , čo je každopádne polynomiálne.
* Riadok 3 potrebuje času, riadky 4 a 5 – polynomiálne.
* Všetky akcie vnútri cyklu (r. 7-15) potrebujú času. Nakoľko medzi každými dvoma vrcholmi vedú nanajvýš dve hrany (tam a späť), tak . Preto riadky 8 a 9 trvajú .
* Cyklus sa preruší, ak algoritmus vyberie hranu, ktorá už je v množine used. Môže sa teda vybrať maximálne toľko hrán, kým ich used nebude obsahovať všetky. Cyklus sa teda vykoná max. krát. Jeho časová zložitosť je teda , čo je polynomiálne.

Všetky kroky algoritmu sú polynomiálne, problém teda patrí do triedy NP.

#### Problém je NP-ťažký:

V tejto časti dôkazu ukážeme redukciu iného NP-úplného problému na náš problém, konkrétne .

Množina a cieľová suma , ktoré sú na vstupe , sa budú transformovať na vstup pre nasledovne:

* Vytvoríme vrchol grafu
* Pre každý prvok z množiny vytvoríme:
  + vrchol (póza )
  + hrany a , obe s váhou (animácia a jej opačný smer)
* Začiatok aj koniec hľadaného ťahu (resp. počiatočná aj konečná póza) bude
* Požadovaná cena ťahu (resp. dĺžka animácie)

Týmto nám vznikne graf v tvare „hviezdičky“ (vzhľadom na obdobie roka môžeme túto hviezdu považovať za Vianočnú) :

1

1

3

3

12

12

25

25

2014

2014

**. . .**

Vrcholov v grafe bude , hrán bude , čiže transformácia je , a teda polynomiálna.

**Dôkaz :**

Vieme, že existuje podmnožina , ktorá sa nasčíta na hodnotu :

Nakoľko váhy hrán , tak v získanom grafe platí:

Výsledný ťah v grafe bude teda postupnosť vrcholov: . Nakoľko je množina, prvky sa v nej neopakujú, a teda aj hrany v  sa nebudú opakovať – je teda naozaj ťah v grafe. Podľa posledného vzťahu tiež vidíme, že cena je rovná . Našli sme teda ťah v grafe, resp. postupnosť animácií , ktorá trvá presne krokov, a teda platí . □

**Dôkaz :**

Vieme, že existuje ťah v grafe získanom z , ktorého cena je , pričom začína aj končí v  . Nakoľko v  sú hrany iba medzi a , neexistuje tam hrana . Tým pádom ani v  neexistuje podpostupnosť takéhoto tvaru, čiže po každom musí nasledovať . Ďalej, graf neobsahuje slučku , teda v  po každom musí nasledovať . Z posledných dvoch tvrdení a faktu, že zo zadania začína aj končí v  nám vyplýva, že:

Nakoľko je ťah, nesmie sa v ňom opakovať žiadna hrana , a teda platí, že všetky sú v  rôzne. Máme teda množinu vrcholov a množinu ich označení . Podľa spôsobu konštrukcie grafu pomocou vieme, že pre všetky vrcholy . Celková cena ťahu je teda rovná:

Nakoľko podľa platí , tak:

Z konštrukcie grafu pomocou vieme, že všetky sú prvkami , a teda . Z predchádzajúceho vzťahu vieme, že súčet tejto podmnožiny je rovný .

Našli sme teda podmnožinu množiny , ktorej súčet je rovný , a teda . □

#### Zhrnutie:

Dokázali sme, že , tiež že , a teda je korektná polynomiálna redukcia. Z toho vyplýva, že .

Tiež sme ukázali polynomiálny nedeterministický algoritmus pre , z čoho vyplýva, že .

Z týchto dvoch záverov vieme odvodiť, že problém je NP-úplný problém.