

# AKO FUNGUJÚ NAVIGAČNÉ SYSTÉMY

Niektorí z vás určite videli navigačný systém v aute, ktorý je po zadaní lokality, kam chcete doraziť, schopný ukázať vám najkratšiu trasu. Tento systém je väčšinou schopný trasu vypočítať veľmi rýchlo napriek tomu, že mapy sú rozsiahle a možných trás je veľmi veľa.

Najprv si vysvetlíme, akým spôsobom má počítač takú mapu uloženú. Na rozdiel od človeka, ktorý keď dostane mapu vo forme obrázka, tak s ňou vie celkom slušne narábať, počítač s takýmto obrázkom priamo pracovať nevie. Potrebuje napríklad zoznam miest a ciest medzi nimi.



Mestá:	Cesty:
Poprad	Poprad–Košice 100 km
Košice	Poprad–Žilina 150 km
Žilina	...
...	

Hore je mapa vhodná pre človeka, dole popis mapy vhodný pre počítač

Naším cieľom je riešiť nasledujúcu úlohu: *Majme mapu cestnej siete, na nej zadané 2 body. Nájdite najkratšiu cestu medzi týmito dvoma bodmi.*

Prvým možným riešením je vyskúšanie všetkých možných trás a z nich vybratie tej najkratšej. Tento prístup má ale jeden háčik. Počet trás vo väčších mapách neuveriteľne rýchlo narastá. To, že v mape, kde je približne 10 miest, by sme skúšali tisícky trás, by sa ešte dalo zvládnuť, avšak pri 20 mestách to už budú milióny a pri 40 miliardy, čo je už priveľa. Nieкто by mohol namietkať, že sme prezerali príliš veľa nerozumných trás, lenže ako počítaču vysvetlite, čo je to nerozumná cesta?

## Zjednodušená úloha

Najprv sa pokúsime vyriešiť jednoduchšiu úlohu. Nebudeme hľadať najkratšiu cestu na základe počtu prejdených kilometrov, ale na základe počtu prejdených ciest medzi mestami (alebo navštívených miest – je to to isté).

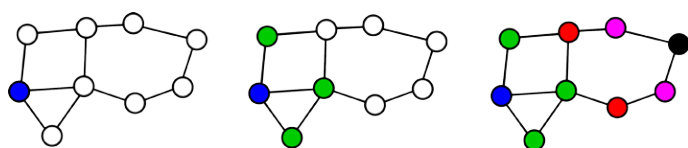
Všimnime si, že ak do mesta B z mesta A vedie cesta dĺžky  $N$ , tak do nejakého zo susedov mesta B vedie cesta z mesta A, ktorá má dĺžku  $N - 1$ . V preklade do bežnej reči: Ak sa viem zo Žiliny do Košíc dostať cestou dĺžky 3 a Košice susedia len s Prešovom a Rožňavou, tak sa musím do jedného z nich vedieť dostať cestou dĺžky 2.

Ak vieme o všetkých mestách, do ktorých sa zo Žiliny vieme dostať cestami dĺžky maximálne 2 (t. j. 0, 1, 2), budeme vedieť nájsť všetky mestá, do ktorých sa budeme vedieť dos-

tať cestou dĺžky 3 – môžu to byť len tie mestá, ktoré susedia s mestami, ktoré majú vzdialenosť 2 od Žiliny.

Zhrňme si celkový postup:

1. Štartovacie mesto má vzdialenosť od seba 0, u ostatných vzdialenosť od štartovacieho miesta nepoznáme.
2. Zoberieme všetkých susedov štartovacieho mesta a ich vzdialenosť určíme ako 1.
3. Zoberieme všetkých susedov miest, ktoré majú vzdialenosť 1 od štartovacieho miesta, a ak nemajú určenú vzdialenosť, tak ju určíme ako 2.
4. Zoberieme všetkých susedov miest, ktoré majú vzdialenosť 3 od štartovacieho miesta, a ak nemajú určenú vzdialenosť, tak ju určíme ako 3.
5. ...



Vľavo: Ukážková mapa cestnej siete. Modrou je označené štartovacie mesto. V strede: Mapa po kroku 2. Zelenou sú označené mestá so vzdialenosťou 1 od štartovacieho miesta. Vpravo: Mapa po skončení. Červenou sú označené mestá so vzdialenosťou 2, fialovou 3 a čiernou 4 od štartovacieho miesta.

To, že tento postup vedie k správnejmu výsledku, sme si už vysvetlili. Ako sme na tom s rýchlosťou? Na každé mesto sa pozrieme práve raz (keďže jeho vzdialenosť sa určí jednoznačne a už sa meniť nebude). Na každú cestu sa pozrieme dvakrát (raz z každého jej konca). To je veľmi dobrý výsledok, lebo dvakrát väčšia mapa bude potrebovať iba dvakrát viac času (na rozdiel od skúšania ciest hrubou silou, kde je nárast obrovský).

Technické detaily pre fajnšmekrov:

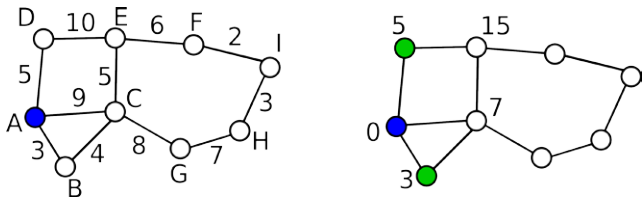
- Počítač by pri vzdialenostiach rád videl čísla, takže pojem neznáma vzdialenosť je nevhodný. Keďže si môžeme byť istí, že žiadne mesto nebude mať zápornú vzdialenosť od štartovacieho miesta, nič nám nebráni použiť vzdialenosť  $-1$  ako neurčenú.
- Aby sme na začiatku krokov 3, 4, ... nemuseli skúmať celú mapu (na nájdenie miest s nejakou vzdialenosťou), je vhodné si mestá, ktorým v danom kroku určujeme vzdialenosť, niekam uložiť. Potom v ďalšom kroku môžeme použiť práve tieto mestá.

## Pôvodná úloha

Teraz sa vráťme k úlohe, v ktorej majú cesty danú dĺžku. Prvé riešenie by bolo rozdeliť cesty nejednotkovej dĺžky imaginárnymi mestami na cesty dĺžky 1 a riešiť prechádzajúcu úlohu. Nie je to úplne zlé riešenie, ale nie je ani najrýchlejšie.

Pozrime sa na postup v zjednodušenom prípade. Poznáme mestá, ktorých vzdialenosť od štartu je menšia ako nejaké číslo  $N$ , a podľa tejto vzdialenosti určujeme vzdialenosť miest, s ktorými susedia, od štartu. Čo by sa stalo v tomto prípade? Tiež by sme o nejakých mestách vedeli, akú majú vzdialenosť od štartu (nazvime túto vzdialenosť definitív-

nou). Pozrieme sa na ich susedov a určíme vzdialenosti od štartu (nazvime ich dočasné). Ak sa do niektorého zo susedov dá dostať viacerými spôsobmi, priradíme mu najkratšiu z možných vzdialeností. Tak sme niečo zistili o susedoch. Ale treba si uvedomiť, že na rozdiel od predchádzajúceho prípadu sme nenašli všade ideálnu vzdialenosť. Na obrázku nižšie vidíme, že niektoré vzdialenosti sa dajú vylepšiť. Ale tá najmenšia možná vzdialenosť sa už vylepšiť nedá, lebo pokiaľ by lepšia trasa šla iba cez mestá, ktoré už majú definitívnu vzdialenosť, tak by sme o nej vedeli. Pokiaľ by šla cez nejaké mesto s dočasnou vzdialenosťou, tak táto by musela byť nižšia, čo opäť nenastane.

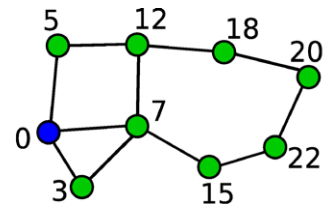
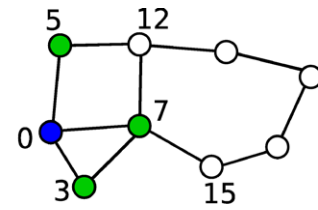
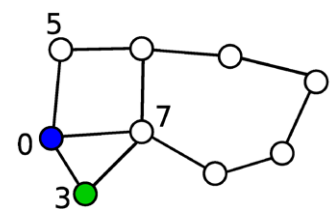
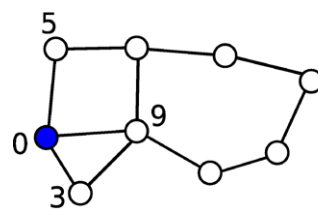


*VLAVO: Príklad cestnej siete, pri cestách sú uvedené ich dĺžky. Modrou je vyznačené štartovacie mesto. Vpravo: Zelenou sú označené mestá, ktorých vzdialenosť máme určenú. A z týchto údajov je dopočítaná vzdialenosť susedov. Všimnite si, že v meste C, ktoré má vzdialenosť od štartu 7, už tento údaj nevylepšíme. Ale v meste E, ktoré má vzdialenosť od štartu 15, dokážeme tento údaj vylepšiť na 12.*

Celý postup môžeme zhrnúť do nasledujúcich bodov. Pod vzdialenosťou budeme myslieť vzdialenosť od štartovacieho miesta.

1. Štartovacie mesto má dočasnú vzdialenosť 0, u ostatných vzdialenosť nepoznáme.
2. Vyberieme mesto s najnižšou dočasnou vzdialenosťou a túto vzdialenosť prehlásime za definitívnu.
3. Preskúmame všetkých jeho susedov a ak sa dá, tak vylepšíme ich dočasné vzdialenosti.
4. Ak sú ešte nejaké mestá, ktoré majú dočasnú vzdialenosť, tak prejdeme na krok 2.

Ako sme na tom s rýchlosťou? Každou cestou opäť prejdeme dvakrát (z každého konca raz). Každé mesto bude v kroku 2 vybrané práve raz. Pri výbere v kroku 2 však musíme skontrolovať všetky mestá, aby sme našli to, ktoré má naj-



*VLAVO hore: Stav po prehľadani susedov štartovacieho mesta. Vpravo hore: Stav po prehľadani susedov mesta B. VLAVO dole: Stav po prehľadani susedov ďalších 2 miest. Vpravo dole: Konečný stav.*

menšiu vzdialenosť od začiatku. Ak máme  $N$  miest, tak krok 2 vykonáme  $N$ -krát a každý takýto krok prehľadá  $N$  miest, čo vedie k  $N^2$  operáciám. Tento krok sa dá ešte trochu vylepšiť.

### Niekoľko poznámok na záver

Algoritmus, ktorým sme hľadali najkratšiu cestu bez ohľadu na dĺžku hrán, nazývame prehľadávanie do šírky (angl. Breadth-first search). Na hľadanie najkratšej cesty pre hrany, ktoré majú nejakú nezápornú dĺžku, sme použili Dijkstrov algoritmus.

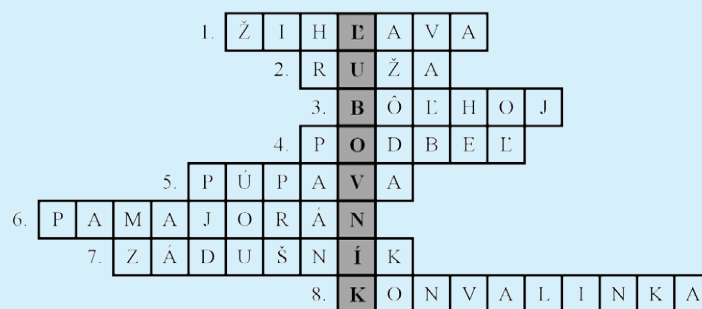
Niekedy nás ani tak netrápi najkratšia cesta, čo sa týka vzdialenosti, ale skôr najkratšia cesta, čo sa týka času. V tomto prípade nám ale nič nebráni miesto kilometrov k cestám napísať koľko času strávime na jednotlivých cestách a potom minimalizovať tento čas.

Taktiež spomínané algoritmy nie sú jediné. Ďalším zaujímavým algoritmom je  $A^*$ , ktorý využíva vopred vypočítaný odhad vzdialenosti (napr. vzdušnú vzdialenosť), na základe ktorého odhaduje, ktorý vrchol prehľadá ako ďalší.

Vladimír Boža

## VYHODNOTENIE KRÍŽOVKY Z 8. ČÍSLA

Správne riešenie krížovky z 8. čísla časopisu bolo **Ľubovník**:



Toto riešenie nám zaslalo 8 čitateľov, z ktorých sme vyžrebovali troch výhercov trička a pier. Tričko a pero časopisu Mladý vedec získava Jana Vajsová z Topoľovky, perá Monika Kmeťová a Mária Kranecová z Modry.

Martin Hriňák