INFORMATIKA

Mosty
(Úlohy z MO kategorie P, 36. část)

PAVEL TÖPFER

Matematicko-fyzikální fakulta UK, Praha

V tomto článku se budeme věnovat zajímavé grafové úloze, kterou jsme již vícekrát využili jako přípravnou úlohu na přednáškách pro řešitele Matematické olympiády kategorie P. Přesně v této podobě nikdy nebyla soutěžní úlohou, vznikla ovšem úpravou a rozšířením jedné praktické soutěžní úlohy z ústředního kola 47. ročníku MO-P (školní rok 1997/98). Původní soutěžní úloha nám také posloužila jako námět úplně prvního dílu našeho dlouhodobého seriálu o úlohách kategorie P matematické olympiády [1], který začal vycházet v časopise Matematika – fyzika – informatika již v roce 2000.

* * * * * * * * * * * *

V jisté zemi mají N měst označených celými čísly od 1 do N. Mezi městy je vybudována silniční síť tvořená M obousměrnými silnicemi. Každá silnice spojuje vždy dvojici měst a je známa její délka. Všechna případná křížení silnic jsou mimoúrovňová, takže mimo města nelze přejet z jedné silnice na druhou. Některé silnice jsou vedeny přes mosty, které mají omezenou nosnost. Pro takové silnice je stanovena maximální povolená hmotnost vozidla, jaké může po silnici projet. Mezi některými dvojicemi měst přímá silnice nevede, ale z každého města lze dojet po silnicích do libovolného jiného města. Mezi každými dvěma městy vede nejvýše jedna přímá silnice. Potřebujeme převézt náklad autem z města A do města B.

a) Zjistěte délku nejkratší cesty z města A do města B. Určete, přes která města tato cesta vede. Pokud existuje více různých cest téže minimální délky, zvolte z nich jednu libovolnou.

b) Zjistěte hmotnost nejtěžšího možného vozidla s nákladem, jaké může přejet po silnicích z města A do města B. Určete, přes která města povede cesta takového vozidla. Pokud existuje více různých cest, po nichž lze přepravit maximální náklad, zvolte z nich jednu libovolnou.

c) Zjistěte délku nejkratší cesty z města A do města B. Určete, přes která města tato cesta vede. Pokud existuje více různých cest téže minimální délky, vyberte z nich tu, po které můžeme převézt co nejtěžší náklad. Existuje-li více takových cest, zvolte z nich jednu libovolnou. Určete také maximální hmotnost vozidla s nákladem, jaké může projet po nejkratší cestě z města A do města B.

d) Zjistěte hmotnost nejtěžšího možného vozidla s nákladem, jaké může přejet po silnicích z města A do města B. Určete, přes která města povede cesta tohoto vozidla. Pokud existuje více různých cest, po nichž lze přepravit maximální náklad, vyberte z nich tu nejkratší. Existuje-li více takových cest, zvolte z nich jednu libovolnou. Určete také délku nejkratší cesty z města A do města B, po níž lze převézt maximální možný náklad.

e) Jak se změní řešení úloh a), b), c), d), když připustíme, že mezi některými dvojicemi měst vede více přímých silnic s různým ohodnocením (silnice jsou různě dlouhé, resp. mají různá omezení maximální povolené hmotnosti vozidla)? Může se tedy třeba stát, že mezi jistými městy X a Y vede jednak krátká přímá silnice, na níž je ale značně omezena hmotnost vozidel, a zároveň jiná delší silnice, kde ale žádné omezení hmotnosti vozidel není.

Vstupní data:

Na prvním řádku vstupu jsou zadána čtyři kladná celá čísla N, M, A, B. První z nich N udává počet měst, předpokládejte $N < 1\,000$. Druhé číslo M určuje celkový počet silnic. Následující čísla A, B představují výchozí a cílové město hledané trasy – jsou to celá čísla z rozmezí od 1 do N. Na dalších M řádcích vstupu jsou zadány informace o jednotlivých silnicích. Každý z těchto řádků je tvořen čtyřmi kladnými celými čísly popisujícími vždy jednu silnici: první dvě z nich jsou čísla měst, mezi nimiž silnice vede (čísla z rozmezí od 1 do N), třetí číslo udává délku silnice v kilometrech (celé číslo z rozmezí od 1 do 100) a čtvrté maximální povolenou hmotnost vozidla v kilogramech (celé číslo z rozmezí od 1 do 10 000). Je-li čtvrtou hodnotou na řádku 0, znamená to, že hmotnost vozidel na této silnici není omezena.

Na první pohled je zřejmé, že se jedná o grafovou úlohu. Silniční síť představuje souvislý neorientovaný graf s ohodnocenými hranami. Každá hrana grafu má přiřazeny dvě hodnoty – délku silnice a případné omezení maximální přípustné hmotnosti vozidel. Pro jednoduchost si můžeme představit, že omezení hmotnosti vozidel je stanoveno pro každou hranu a že má hodnotu „nekonečno“, jestliže na příslušné silnici žádný most není. V základním zadání úlohy graf neobsahuje žádné mutihrany (tj. více hran spojujících stejnou dvojici vrcholů). Na závěr se v podúloze e) ještě zamyslíme, co by se na našem řešení změnilo, kdybychom multihrany připustili.

Zadaný graf si můžeme reprezentovat nejlépe pomocí seznamů následníků jednotlivých vrcholů. Jedná se o neorientovaný graf, takže přímou silnici mezi městy X, Y si musíme uložit dvakrát: vrchol X bude zařazen do seznamu následníků vrcholu Y a také naopak vrchol Y bude zařazen do seznamu následníků vrcholu X. Silnice mají dvoje ohodnocení – délku a maximální přípustnou hmotnost vozidel. V každém záznamu v seznamu následníků proto budou zapsány také tyto dvě hodnoty.

Jinou možností uložení grafu by byla matice vzdáleností. To je matice velikosti $N \times N$, jejíž prvky obsahují ohodnocení jednotlivých hran. V našem případě by tedy každý prvek matice obsahoval dvě hodnoty, a to délku silnice a její hmotnostní omezení. Tato reprezentace grafu je ovšem pro naši úlohu méně vhodná. Představuje větší paměťové nároky i pomalejší výpočet – kdykoliv potřebujeme najít všechny sousedy zvoleného vrcholu, museli bychom v matici projít celý řádek délky N. Navíc bychom v části e) ještě narazili na technický problém, jak si uložit multihrany. Tuto variantu reprezentace grafu v programu proto raději nepoužijeme.

Úloha a)

V této části úlohy chceme nalézt nejkratší cestu z města A do města B bez ohledu na případné omezení hmotnosti vozidel. Všechna ohodnocení hran grafu jsou kladná, takže pro hledání nejkratší cesty v grafu mezi danými dvěma vrcholy můžeme využít dobře známý Dijkstrův algoritmus. Připomeňme si ve stručnosti, jak Dijkstrův algoritmus pracuje (podrobnější popis včetně programu najdete například v učebnici [2] nebo na internetu v textu [3]). Pro každý vrchol U budeme počítat jeho hodnotu, která udává délku dosud nejkratší nalezené cesty z počátečního vrcholu A do příslušného vrcholu U. Navíc budeme evidovat, zda je tato hodnota zatím jenom dočasná (možná ji ještě zlepšíme, tzn. snížíme), nebo zda je již trvalá. Na začátku výpočtu má vrchol A má hodnotu 0 (jsme tam,

nemusíme nikam chodit) a všechny ostatní vrcholy grafu mají hodnotu „nekonečno“ (zatím jsme do nich žádnou cestou nedorazili). Všechny vrcholy mají dočasnou hodnotu.

Vlastní výpočet se odehrává v postupných krocích. V každém kroku výpočtu nejprve najdeme vrchol X s nejmenší dočasnou hodnotou a tuto hodnotu prohlásíme za trvalou. To si můžeme dovolit díky nezápornému ohodnocení hran, v grafu se záporně ohodnocenými hranami by toto nemuselo platit. Následně všem vrcholům sousedícím s vrcholem X, které zatím mají dočasnou hodnotu, zkusíme tuto jejich dočasnou hodnotu zlepšit. Pokud se do takového vrcholu Y dostaneme přes vrchol X kratší cestou, než jakou nejlepší cestu z A do Y jsme dosud našli, pak hodnotu vrcholu Y patřičně snížíme. Novou hodnotou vrcholu Y se v tom případě stane součet hodnoty vrcholu X a délky hrany XY. Zároveň si poznamenáme, že nejlepší hodnotu vrcholu Y jsme získali pomocí vrcholu X, neboli že předchůdcem vrcholu Y na nejkratší cestě z A do Y je právě vrchol X. Výpočet probíhá tak dlouho, dokud nezíská cílový vrchol B trvalou hodnotu. Ta potom určuje délku nejkratší cesty z A do B.

Po určení délky nejkratší cesty z A do B zbývá ještě zjistit, kudy tato cesta vede. K tomu nám poslouží uložené informace o tom, kdo je předchůdcem kterého vrcholu na nejkratší cestě. Celou cestu zrekonstruujeme odzadu tzv. zpětným chodem. Začneme od cílového vrcholu B, ten zná svého předchůdce, ten zase svého předchůdce atd. Takto postupujeme, dokud nedojdeme až do počátečního vrcholu A. Pokud z vrcholu A do vrcholu B existuje více různých cest téže minimální délky, pak popsaný algoritmus vyhledá tu z nich, kterou našel jako první.

Časová složitost závisí na způsobu implementace. Výpočet nejkratších vzdáleností vyžaduje provést v nejhorším případě až N kroků, pokud koncový vrchol B získá trvalou hodnotu až jako poslední. V každém kroku výpočtu představuje nalezení vrcholu s minimální dočasnou hodnotou práci $O(N)$, jsou-li hodnoty vrcholů uloženy jednoduše v poli, které musíme sekvenčně procházet. Následné přepočítání hodnot sousedních vrcholů zvládneme jistě také v čase $O(N)$. To vede k hornímu odhadu časové složitosti celého algoritmu $O(N^2)$. Zpětný chod je pak už jednoduchý průchod nejvýše N vrcholy s časovou složitostí $O(N)$. V odborné literatuře se můžete dočíst o časově výhodnější implementaci Dijkstrova algoritmu, pokud si hodnoty vrcholů ukládáme například do haldy. Možnostem takové časové optimalizace se v našem článku nebudeme věnovat, jsou uvedeny třeba ve studijním textu [3].

Úloha b)

K řešení úlohy b) můžeme přistoupit různými způsoby. Jednou možností je vyjít z pozorování, že hmotnost nejtěžšího možného vozidla s nákladem, jaké může přejet po silnicích z města A do města B, bude jistě rovna hmotnostnímu limitu některé z existujících silnic. Takových různých hodnot je nejvýše M. Pro každou z těchto hodnot stačí ověřit, zda vozidlo příslušné hmotnosti dokáže přejet po silnicích z města A do města B, neboli zda vrcholy A, B leží ve stejné komponentě souvislosti, jestliže z grafu vynecháme všechny hrany s nižším hmotnostním limitem. Z vyhovujících hodnot vezmeme tu nejvyšší. Tím máme určenu optimální hmotnost vozidla a pro takové vozidlo pak již snadno najdeme nějakou cestu z města A do města B po vyhovujících silnicích například prohledáváním grafu do šířky. Časová složitost tohoto algoritmu je $O(M(M+N))$. Uvažujeme až M různých hmotností vozidla a pro každou z nich určujeme komponentu souvislosti grafu obsahující vrchol A, což představuje práci $O(M+N)$. Závěrečné vyhledání cesty z A do B prohledáváním do šířky má časovou složitost $O(M+N)$, což výslednou asymptotickou složitost celého řešení $O(M(M+N))$ nijak neovlivní.

Výpočet můžeme urychlit, pokud nebudeme zkoušet všech M potenciálních hodnot hmotnosti vozidla. Tyto hodnoty můžeme v čase $O(M \log M)$ uspořádat v poli a nejvyšší použitelnou pak mezi nimi vyhledat metodou půlení intervalů. To vyžaduje provést pouze $\log M$ kroků, v nichž stejně jako dosud testujeme, zda vrcholy A, B leží v téže komponentě souvislosti grafu. Asymptotická časová složitost celého výpočtu se tím sníží na $O(M \log M + (M+N) \log M)$, což můžeme zjednodušit na $O((M+N) \log M)$. V grafu bez multihran můžeme celkový počet hran M shora odhadnout pomocí N^2, takže odhad výsledné časové složitosti můžeme zapsat také jako $O((M+N) \log N)$.

Ukážeme si ještě jiný způsob řešení. Úloha b) vypadá na první pohled velmi podobně jako úloha a), takže se nabízí také ji řešit obdobným způsobem a použít k řešení Dijkstrův algoritmus. Musíme ale navrhnout jeho vhodnou modifikaci, jelikož s hodnotami vrcholů grafu nyní budeme počítat trochu odlišným způsobem. Každému vrcholu V našeho grafu přiřadíme hodnotu, která bude určovat nejvyšší hmotnost vozidla, jaké jsme zatím dokázali dopravit z města A do města V. Opět si zavedeme evidenci, zda je tato hodnota dočasná (mohla by se ještě zlepšit, tzn. zvýšit), nebo už trvalá. Na začátku výpočtu mají všechny vrcholy dočasnou hodnotu – vrchol A má hodnotu „nekonečno“ (jsme tam s jakýmkoliv vozidlem bez

omezení hmotnosti) a všechny ostatní vrcholy grafu mají hodnotu 0 (zatím jsme do nich nedojeli s žádným nákladem).

V každém kroku výpočtu najdeme vrchol X s největší dočasnou hodnotou a tu označíme za trvalou. Poté všem vrcholům sousedícím s vrcholem X, které zatím mají dočasnou hodnotu, zkusíme tuto jejich dočasnou hodnotu zlepšit. Pokud se do takového vrcholu Y dostaneme přes vrchol X s těžším vozidlem, než jaké jsme dosud z A do Y dopravili, pak hodnotu vrcholu Y patřičně zvýšíme. Novou hodnotou vrcholu Y se v tom případě stane minimum z hodnoty vrcholu X a hmotnostního limitu hrany XY. Do města Y totiž může dojet po přímé silnici z města X nejvýše takové vozidlo, jaké dojede z A do X a může pak také dále pokračovat po silnici XY. Zároveň si poznamenáme, že nejlepší hodnotu vrcholu Y jsme získali pomocí vrcholu X, neboli že předchůdcem vrcholu Y na nejvýhodnější cestě z A do Y je právě vrchol X. Výpočet probíhá tak dlouho, dokud nezíská cílový vrchol B trvalou hodnotu. Ta potom určuje hmotnost nejtěžšího vozidla, jaké dojede z A do B. Nalezení průběhu optimální cesty zpětným chodem pomocí zaznamenaných předchůdců se oproti klasickému Dijsktrovu algoritmu nijak nezmění.

Vidíme, že průběh popsaného řešení přesně kopíruje Dijsktrův algoritmus z řešení úlohy a), jenom místo výběru minima vybíráme ve vrcholech maximum a místo počítání součtu počítáme minimum. Stejná proto zůstane i asymptotická časová složitost algoritmu – při jednoduché implementaci s hodnotami vrcholů uloženými v poli je to $O(N^2)$.

Úloha c)

Úloha c) přímo navazuje na obě předchozí úlohy a), b). Nyní chceme zvolit nejlepší cestu podle dvou kritérií. Chceme nalézt co nejkratší cestu z města A do města B a existuje-li více takových různých cest téže minimální délky, máme vybrat tu z nich, kudy převezeme co nejtěžší náklad.

Základem řešení bude Dijkstrův algoritmus popsaný v řešení úlohy a). Algoritmus ovšem musíme trochu doplnit, aby z více cest téže optimální délky nebral jednoduše první nalezenou, ale aby se v takovém případě rozhodoval podle druhého kritéria. Tím je co největší hmotnost vozidla, které může projet po nalezené cestě. K úpravě algoritmu dojde pouze v jednom místě. Když hodnotu vrcholu X prohlásíme za trvalou a zkoumáme, zda můžeme zlepšit hodnotu sousedního vrcholu Y, může se stát, že se dosavadní hodnota vrcholu Y přesně rovná součtu hodnoty vrcholu X a délky hrany XY. To znamená, že nyní přicházíme do vrcholu Y z vrcholu X po cestě stejné délky, jakou měla dosud nejlepší nalezená cesta vedoucí

z vrcholu A do vrcholu Y. Z hlediska určení nejkratší cesty z A do B je v takovém případě jedno, zda záznam o předchůdci vrcholu Y ponecháme beze změny, nebo zda ho změníme na X. Ovlivníme tím jenom to, kterou z více možných cest téže minimální délky algoritmus nakonec při zpětném chodu vyhledá (pokud ovšem vůbec bude vrchol Y ležet na nejkratší cestě z A do B). A to je přesně ten okamžik, kdy k rozhodnutí o předchůdci vrcholu Y použijeme druhé kritérium.

Zbývá ukázat, jak popsanou myšlenku zrealizujeme. Hodnotám jednotlivých vrcholů, s nimiž počítá řešení úlohy a), budeme nadále říkat „první hodnota“. Ke každému vrcholu grafu si teď uložíme ještě jeho „druhou hodnotu“, která bude určovat maximální hmotnost vozidla, jaké může do tohoto vrcholu dojet z výchozího vrcholu A – ovšem pouze po dosud nejkratší nalezené cestě!

Kdykoliv přijdeme z vrcholu X do vrcholu Y po nějaké kratší cestě a budeme díky tomu snižovat první hodnotu vrcholu Y a zároveň měnit záznam o předchůdci vrcholu Y, přepočítáme zároveň i druhou hodnotu vrcholu Y. Novou druhou hodnotou vrcholu Y se stane minimum z druhé hodnoty vrcholu X a hmotnostního omezení hrany XY. Do vrcholu Y může totiž přijet z vrcholu X nejvýše tak těžké vozidlo, jaké může dojet ze startu A do vrcholu X a následně může také projet po přímé silnici z X do Y. Je to úplně stejný způsob výpočtu, jaký jsme použili již v řešení úlohy b). Poznamenejme, že druhá hodnota vrcholu Y se tím může i snížit oproti stávajícímu stavu, tedy může se zhoršit. K tomu dojde v případě, že jsme našli kratší cestu z A do Y, která ale připouští pouze lehčí vozidla.

Kdykoliv přijdeme z vrcholu X do vrcholu Y po cestě stejné délky, jakou jsme našli již dříve, rozhodneme se o případné změně předchůdce vrcholu Y podle druhého kritéria. Spočítáme si minimum z druhé hodnoty vrcholu X a z hmotnostního omezení hrany XY. Toto číslo porovnáme s dosavadní druhou hodnotou vrcholu Y. Je-li spočtené číslo větší, znamená to, že jsme se právě dostali do vrcholu Y po cestě sice stejné délky, ale s těžším vozidlem. To je pro nás zlepšení. V takovém případě proto druhou hodnotu vrcholu Y patřičně zvýšíme a záznam o nejlepším předchůdci vrcholu Y změníme na X.

Na vyhledání cesty zpětným chodem podle zaznamenaných předchůdců se pak už nic nemění. Provedenou úpravou se nijak nezmění ani asymptotická časová složitost algoritmu, stejně jako v úloze a) bude rovna $O(N^2)$. Podrobnější popis jiné varianty Dijkstrova algoritmu s více kritérii včetně programové realizace si můžete přečíst v učebnici [2].

Úloha d)

Úloha d) vypadá velmi podobně jako úloha c), prohazuje pouze prioritu obou kritérií při výběru cesty. Chceme nalézt cestu z A do B umožňující převezení co nejtěžšího nákladu a z více takových cest zvolit tu nejkratší. Mohlo by se tedy zdát, že řešení obou těchto úloh budou prakticky stejná, jenom se prohodí pořadí obou použitých kritérií. Tak by tomu skutečně bylo v případě jiné dvojice uvažovaných kritérií, například kdybychom pro každou silnici znali vedle její délky ještě očekávanou dobu jízdy, chtěli bychom nalézt nejkratší cestu z města A do města B a z více takových stejně dobrých možností zvolit tu s nejkratší dobou jízdy. Doby jízdy v jednotlivých navazujících úsecích cesty se totiž stejně jako délky cest jednoduše sčítají. V zadané úloze ale pracujeme s maximální povolenou hmotností vozidla a v tomto případě se namísto součtu počítá minimum. Tato zdánlivá maličkost vede k tomu, že pouhé prohození pořadí obou kritérií výběru nefunguje. Ukážeme si to nejlépe na jednoduchém příkladu.

Použijeme Dijkstrův algoritmus se dvěma kritérii, kde prvním kritériem nyní bude maximální přípustná hmotnost vozidla a druhým minimální délka cesty. Algoritmus se tedy snaží do každého vrcholu především dopravit z počátečního vrcholu A co nejtěžší vozidlo a pokud toho lze dosáhnout více způsoby, zvolí z nich tu cestu, která je co nejkratší. Podle toho budeme v jednotlivých vrcholech průběžně přepisovat záznamy o předchůdci na nejlepší cestě obdobným způsobem, jako jsme to dělali v řešení úlohy c). Předpokládejme, že z vrcholu A vede přímá silnice do vrcholu Y délky 10 km s omezením hmotnosti vozidel do 5 000 kg. Dále je možné jet z vrcholu A do vrcholu Y oklikou přes vrchol X po cestě o celkové délce 20 km, kudy ale mohou projet vozidla o hmotnosti až 8 000 kg. Konečně z vrcholu Y vede přímá silnice do vrcholu B délky 5 km s omezením hmotnosti vozidel do 3 000 kg. Jiné silnice v grafu nejsou. Uvažovaný algoritmus označí za trvalé vrcholy v pořadí A, X, Y, B. Při zpracování vrcholu Y zvolí za výhodnější tu cestu, která vede přes vrchol X, neboť tato cesta umožňuje dopravit do vrcholu Y těžší náklad (8 000 kg), než přímá silnice z A do Y (5 000 kg). Předchůdcem vrcholu Y na nejlepší cestě se proto definitivně stane vrchol X. V té době algoritmus ještě netuší, že vzhledem k hmotnostnímu omezení následující silnice YB se stejně do cíle dostane nejvýše vozidlo o hmotnosti 3 000 kg. Cílový vrchol B tak nakonec dostane správnou výslednou hodnotu 3 000 kg, ale při výpisu optimální cesty zpětným chodem pomocí zaznamenaných předchůdců dostaneme chybně cestu A-X-Y-B (25 km), ačkoliv cesta A-Y-B také umožňuje projet vozidlu o hmotnosti 3 000 kg a je navíc kratší (15 km).

Vidíme, že Dijkstrův algoritmus se dvěma kritérii v tomto případě nelze použít a k řešení budeme muset přistoupit jiným způsobem. V první fázi určíme pouze hmotnost nejtěžšího vozidla, jaké může dojet z vrcholu A do vrcholu B (označme ji V). Hodnotu V získáme kterýmkoliv ze způsobů uvedených v řešení úlohy b). Můžeme použít třeba výše uvedenou modifikaci Dijkstrova algoritmu, kde se místo součtů budou počítat minima a v každém kroku se bude uzavírat vrchol s dosud maximální, nikoliv minimální dočasnou hodnotou. Výsledkem této fáze výpočtu je hmotnost V, nebudeme zatím hledat průběh cesty zpětným chodem. Ve druhé fázi pak chceme nalézt nejkratší cestu vozidla o hmotnosti V z města A do města B. Přitom již víme, že taková cesta existuje. Z grafu můžeme vypustit všechny hrany s hmotnostním limitem menším než V, tzn. nebudeme uvažovat ty silnice, po nichž nesmí projet vozidlo této váhy. V takto upraveném grafu najdeme nejkratší cestu z A do B použitím standardního Dijkstrova algoritmu, tzn. stejně jako v řešení úlohy a).

Obě fáze výpočtu mají při jednoduché implementaci s hodnotami uloženými v poli asymptotickou časovou složitost $O(N^2)$, vypuštění zakázaných hran mezi oběma fázemi jistě zvládneme v čase $O(M)$, což lze v grafu bez multihran shora odhadnout také pomocí $O(N^2)$. Celková časová složitost celého řešení je tudíž opět rovna $O(N^2)$.

Úloha e)

Dosud jsme neuvažovali možnost, že by mezi některými dvojicemi měst vedlo více silnic s různými parametry, tedy že by v grafu existovaly multihrany. Pokud takovou možnost připustíme, v řešení úlohy a) se vůbec nic nezmění. Jediným kritériem je zde pro nás délka zvolené cesty. Kdyby mezi dvojicí měst X, Y vedly dvě silnice různé délky, delší z nich se ve výsledné cestě zcela jistě neuplatní. Kdyby takové dvě silnice XY měly stejnou délku, pak stačí uvažovat jednu libovolnou z nich. Problém multihran tedy snadno vyřešíme již při čtení vstupních dat, kdy si pro každou dvojici měst zapamatujeme pouze jednu ze spojujících silnic – vždy tu nejkratší.

Stejným způsobem vyřešíme případné multihrany také v úloze b). Zde nás na silnicích zajímají pouze omezení hmotnosti vozidel. Z více přímých silnic spojujících dvojicí měst X, Y stačí při hledání cesty co nejtěžšího vozidla uvažovat pouze jednu – tu, která je nejvýhodnější, která připouští co největší hmotnost projíždějících vozidel. Všechny ostatní silnice můžeme opět vynechat již při čtení vstupních dat.

Podobná situace nastává i při řešení úlohy c), v níž pracujeme již se dvěma výběrovými kritérii. Prioritním kritériem je zde pro nás délka zvolené cesty. Proto stejně jako v úloze a) platí, že kdyby mezi dvojicí měst X, Y vedly dvě silnice různé délky, delší z nich se ve výsledné cestě zcela jistě nepoužije. Pokud by takové dvě silnice měly stejnou délku, ale lišily by se v omezení hmotnosti vozidla, stačí zase uvažovat jenom tu z nich, která připouští průjezd těžšího vozidla. Problém multihran tak opět vyřešíme při čtení vstupních dat, kdy si pro každou dvojici měst uložíme pouze jednu nejvýhodnější ze spojujících hran (co nejkratší, příp. z více takových tu s nejvyšším hmotnostním limitem).

V úloze d) je ovšem situace odlišná. Pokud mezi městy X, Y vedou dvě přímé silnice, z nichž první umožňuje převézt těžší náklad, ale druhá je kratší, nemůžeme předem vyloučit ani jednu z nich. Nyní má při výběru cesty přednost maximální povolená hmotnost vozidla, takže by se zdálo, že vždy dáme přednost té první silnici. Hmotnostní omezení platná na silnicích v ostatních částech výsledné cesty mohou však způsobit, že stejně nemůžeme z města A do města B žádným způsobem dovézt těžší náklad, než kolik připouští druhá z uvažovaných silnic mezi městy X, Y. Přitom volba této druhé silnice povede k celkově kratší cestě z A do B, takže součástí optimální cesty ve skutečnosti bude druhá silnice. Popsanou situaci dobře demonstruje jednoduchý příklad uvedený v řešení úlohy d). V tomto případě se tedy nemůžeme předem zbavit multihran již při čtení vstupních dat, jako jsme to udělali v předchozích třech úlohách. Musíme si všechny přímé silnice uložit do vhodné datové struktury reprezentující naši silniční síť a pracovat s nimi při hledání výsledné cesty. Vypustit bychom mohli jedině takovou přímou silnici mezi městy X, Y, která by byla horší z obou uvažovaných hledisek ve srovnání s nějakou jinou přímou silnicí spojující města X, Y (tzn. zrušit můžeme pouze takovou silnici, která je delší a zároveň také povoluje průjezd pouze pro lehčí vozidla).

Na závěr pro úplnost dodejme, že v původní soutěžní úloze 47. ročníku Matematické olympiády kategorie P se řešila pouze jistá varianta naší podúlohy d). Na silnicích se místo mostů nacházely tunely, které omezovaly maximální možnou výšku vozidel. To je ovšem změna pouze kosmetická, která na principu řešení vůbec nic nemění. Druhou odlišností bylo, že nebyly zadány délky jednotlivých silnic. Místo hledání nejkratší cesty pro nejvyšší možné vozidlo se hledala taková cesta, kudy nejvyšší možné vozidlo projede z výchozího do cílového města přes co nejmenší počet měst. Hledanou cestu v neohodnoceném grafu tak bylo možné získat prohledává-

ním grafu do šířky, zatímco my jsme v naší úloze pracovali s ohodnoceným grafem a využili jsme proto Dijkstrův algoritmus. Zadání i řešení původní soutěžní úlohy uvádí internetový archív úloh MO kat. P uložený na adrese [4] a podrobněji ho popisuje článek [1].

Literatura

[1] *Töpfer, P.*: Úlohy z MO – kategorie P. MFI, roč. 9 (1999–2000), č. 6, s. 375–379.

[2] *Töpfer, P.*: Algoritmy a programovací techniky. Prometheus, Praha, 1995 a 2007.

[3] http://ksp.mff.cuni.cz/kucharky/halda-a-cesty/

[4] http://mo.mff.cuni.cz/p/archiv.html

Jak počítal první programovatelný počítač?

INGRID NAGYOVÁ

Pedagogická fakulta OU, Ostrava

Prvenství v konstrukci programovatelného počítače je přisuzováno německému inženýrovi Konrádovi Zuse, který v roce 1938 sestrojil první elektromechanický stroj. Tento stroj s názvem Z1 pracoval s čísly ve dvojkové soustavě a byl ovladatelný pomocí programu zadávaného na děrné pásce. Počítač Z1 byl velmi poruchový a proto nevhodný pro praktické využití. Odstartoval však vývoj dalších, daleko sofistikovanějších počítacích strojů a vedl nakonec ke konstrukci počítače tak, jak jej známe dnes. Počítače Konráda Zuse byly využity ve válce a před koncem války byly zničeny při náletu. Archiv prací Konráda Zuse lze dnes najít i na internetu [1]. K záznamům o konstrukci počítačů se o několik desetiletí později vrátil profesor Raúl Rojas, který ukázal, že počítače Z1 a Z3 jsou i přes absenci podmíněných skoků Turingovsky úplné. Zaměřil se také na rekonstrukci počítacích strojů Konráda Zuse [2]. Replika počítače Z1 se dnes nachází v Technickém muzeu v Berlíně. Panoramatickou vizualizaci počítače a simulaci výpočetního procesu lze také sledovat na [3].