

Semestrální práce

Téma: **Budget Travel**

Vypracoval: **Marek Handl**

Datum: květen 2006

Cvičení: pondělí, 9:15, Vít Rokos

Zadání:

Oficiální znění zadání (v angličtině) je zde: <http://acm.uva.es/p/v2/222.html>

Úkolem je najít pro zákazníka nejlevnější cestu ze startovního do cílového města. Situaci si lze představit tak, že existuje pouze jedna silnice mezi těmito městy a na ní jsou v různých vzdálenostech od startu rozmístěny benzínové stanice (maximální počet je 50). U každé benzínky známe její vzdálenost od startu a cenu benzínu za galon. Dále známe vzdálenost cílového města, kapacitu benzínové nádrže automobilu a spotřebu (kolik milí se ujede na galon).

Zákazník se chová podle těchto pravidel:

- ve startovním městě naplní celou nádrž za cenu, která je zadána
- pokud to není nutné, tak nikdy netankuje, pokud nemá nádrž prázdnou alespoň z poloviny
- pokud tankuje, tak vždy nabere plnou nádrž
- pokud tankuje, tak utratí další \$2 za občerstvení
- stačí mu dojet do cílového města, více benzínu nepotřebuje
- platba na benzínce se zaokrouhluje na centy

Výstupem má být minimální částka, kterou zákazník zaplatí při cestě ze startu do cíle.

Formát vstupního souboru:

Vstupní soubor je textový soubor obsahující reálná čísla. Benzínky jsou seřazeny podle vzdálenosti od nejmenší k největší. Soubor se skládá z několika bloků s tímto uspořádáním:

```
vzdálenost_do_cílového_města
kapacita_nádrže spotřeba cena_za_nádrž_na_startu počet_benzínek
vzdálenost_benzínky cena_benzínu
vzdálenost_benzínky cena_benzínu
...
...
...
vzdálenost_benzínky cena_benzínu
```

Za posledním blokem je uvedeno záporné číslo.

Popis řešení:

Problém lze popsat tak, že máme orientovaný graf, jehož vrcholy jsou start, cíl a benzínky. Hrany vedou ve směru od startu (každý vrchol má svou vzdálenost) a hrana z vrcholu u do vrcholu v vede právě tehdy, pokud platí $(v.vzdálenost - u.vzdálenost) \leq \text{dosah}$, kde dosah je počet milí, které auto ujede na plnou nádrž. Hrany jsou ohodnoceny takto – nechť hrana h vede z u do v , potom $h.\text{hodnoceni} = (v.vzdálenost - u.vzdálenost) \cdot v.\text{cena_benzinu} + 2$. Hrany končící v cíli mají ohodnocení rovné nule. Potom stačí najít nejkratší cestu (vzhledem k ohodnocení hran) ze startu do cíle.

K hledání cesty jsem vytvořil algoritmus vycházející z Dijkstrova algoritmu.

Jelikož počet vrcholů grafu je malý (maximálně 52), použil jsem pro reprezentaci grafu statické

pole vrcholů. Každý vrchol je reprezentován strukturou nesoucí informace o vzdálenosti vrcholu od startu, ceně benzínu v daném vrcholu, ceně doposud nalezené nejkratší cesty do vrcholu a o uzavřenosti vrcholu (poté, co je vrchol uzavřen, již nebude měněna cena cesty do něj). V poli jsou vrcholy seřazeny podle vzdálenosti od startu. Hrany není třeba nijak zaznamenávat, protože je lze vypočítat z vlastností vrcholů.

Na začátku všem vrcholům přiřadím cenu nekonečno, startovnímu uzlu přiřadím cenu za nádrž ve startovním městě. Všechny vrcholy jsou otevřené. Začnu od prvního vrcholu (startu), jeho následovníkům přiřadím cenu (pokud je nižší než dříve zaznamenaná cena) podle ohodnocení hrany a ceny aktuálního vrcholu. Nejprve zjistím, jestli je mezi následníky i cíl, pokud ano, končím výpočet, cena nejlevnější cesty je cena aktuálního vrcholu. Jinak procházím všechny následníky, kteří jsou dál než, co se ujede na půl nádrže. Pokud žádný takový následník není, projdu i následníky, kteří jsou blíže. Poté uzavřu vrchol. Najdu vrchol s nejmenší cenou (lineární průchod polem) a opakuji postup.

Výstup programu se zapisuje do výstupního souboru.

Složitost:

Počet vrcholů značím n , odpovídá to počtu benzínek (zvětšeno o 2).

Inicializace (příprava pole vrcholů): $\Theta(n)$

Vnější while ve funkci `spocitej` se provede maximálně n -krát (aktuálním vrcholem se může stát pouze otevřený vrchol, v každém průchodu jeden vrchol uzavřu).

Vnitřní while-cykly dohromady provedou $\Theta(n)$ operací.

Nalezení minima má složitost $\Theta(n)$, protože se lineárně prochází pole vrcholů.

Celková složitost tedy je: $\Theta(n) + n \cdot (\Theta(n) + \Theta(n)) = \Theta(n^2)$

Vypočítaná složitost platí pro jednu sadu vstupních dat.

Asymptotického zrychlení by mohlo být dosaženo jinou reprezentací grafu. Hrany by se reprezentovaly seznamem následníků, vrcholy by se řadily haldou. Jelikož ale počet hran v tomto grafu je vysoký, nemá takovéto zlepšení velký význam. Naopak by byl program mnohem složitější a pravděpodobně i pomalejší z důvodu práce s pointerem.

Realizace:

Program jsem napsal v jazyce C pro kompilátor Borland, jeho zdrojový kód je v souboru `budgettr.c`. Překlad a spouštění jsem prováděl pouze na Windows platformě. Při spouštění je nutné zadat z příkazové řádky dva parametry – vstupní a výstupní soubor, tedy takto: `budgettr.exe vstupni_soubor vystupni_soubor`.

Soubory `input*.txt` obsahují testovací data.