Programovací techniky

Standardní zadání semestrální práce pro PT 2020/2021

Jiří Andrlík, K19B0514P, jandrlik@students.zcu.cz
Karolína Motyčková, K19B0524P, kmotycko@students.zcu.cz

Obsah

Zadání

Společnost „Dřevák a syn“ vlastní po celém světě síť D dřevozpracujících továren. Snaha společnosti je minimalizovat cenu přepravy výrobků z továren do supermarketů. Starý Dřevák nikdy nešel příliš s dobou, co se kam má přepravit (samozřejmě aby to vyšlo nejlevněji), vždy počítal na papír. Nyní ale převzal společnost mladý Luboš Dřevák, který si na to chce napsat software a místo celovečerního počítání sledovat TokTik nebo hrát GallFuys přes Diskord s přáteli.

Všechny dřevozpracující továrny dokáží produkovat Z druhů různého zboží, od skříní přes dekorace až po záchodová prkénka. Své výrobky rozváží do S supermarketů, nejčastěji do supermarketů BOBI a PATu, ale ta prkénka chtějí i v CUUPu. Vedení každého supermarketu s je však náročné, zvláště pak v BOBI na Borech (který mají interně zařazený jako pátý v pořadí, tedy s = 5, kde to vypadá, jakoby management generoval poptávku náhodně) a každý den t hodlá daný supermarket prodat jiný počet zboží z (hodnotu poptávaného počtu zboží značíme $r_{s,z,t}$) svým zákazníkům. S továrnami to tak snadné není, protože každá továrna d dokáže vyprodukovat pouze omezený počet produktů daného typu $p_{d,z,t}$ v závislosti na tom, kolik jim dodavatelé daný den zrovna (například kvůli koronaviru) dodají materiálu a také jak se zrovna chce zaměstnancům do práce. V továrně d = 2 v Mostu se navíc poslední dobou často stávkuje. Cena převozu zboží z továrny do supermarketu je dána pouze továrnou a supermarketem (jeden kus libovolného zboží lze převézt mezi dvěma uzli s konstantní cenou $c_{s,d}$, tedy například převoz jednoho záchodového prkénka, dvou matrjošek a dvou skříní z továrny v Mostu do supermarketu BOBI na Borech stojí $5c_{5,2}$). Mladík však převzal společnost po otci náhle (otec mu řekl, že to nedokáže lépe, protože už to dělá léta, ať se tedy ukáže), a v supermarketech tedy zbyly zásoby zboží $q_{s,z}$. Pomozme Lubošovi naplánovat přepravu zboží pro T dní dopředu. Luboš zná:

D – počet továren,

S – počet supermarketů,

Z – počet druhů zboží,

T – počet dní,

$c_{s,d}$ – cenu převozu jednoho kusu libovolného zboží z továrny d do supermarketu s,

$q_{z,s}$ - počáteční zásoby výrobků z v supermarketu s,

$p_{d,z,t}$ – produkci továrnou d druhu zboží z v den t,

$r_{s,z,t}$ – poptávku zákazníků po zboží z ze supermarketu s dne t,

a chce znát $k_{s,d,z,t}$ pro $\forall s \in \{1,\ldots, S\}$, $d \in \{1,\ldots, D\}$, $z \in \{1,\ldots, Z\}$, $t \in \{1,\ldots, T\}$, tedy počty kusů všech druhů zboží z rozvezených ze všech továren d do všech supermarketů s každý den t. Pokud by se stalo, že továrny nedokáží uzásobit supermarkety, Luboš potřebuje znát, kdy to bude (den t), aby doobjednal zboží z Číny. Supermarkety mají povoleno prediktivní plánování s neomezenými zásobami (někteří tomu mohou říkat „křečkování“) – lze tedy předzásobit supermarket dle dat poptávky z „budoucna“. Jestli toho využijete či nikoliv je na Vás, je ale třeba toto uvést do dokumentace (viz dále).

Seznamte se se strukturou vstupních dat (počty, ceny cest, objem poptávek...) a načtěte je do svého programu. Formát souboru je popsán přímo v záhlaví vstupních souborů. Navrhněte vhodné datové struktury pro reprezentaci vstupních dat, zvažujte časovou a paměťovou náročnost algoritmu pracujících s danými strukturami. Proveďte základní simulaci pro první den, vypište celkovou cenu přepravy prvého dne pomocí výrazu $\sum_{s=1}^{S} \sum_{d=1}^{D} (c_{s,d} \sum_{z=1}^{Z} k_{s,d,z,1})$

Vytvořte prostředí pro snadnou obsluhu programu (menu, ošetření vstupů) - nemusí být grafické, umožněte manuální zadání požadavku umožněte sledování (za běhu simulace) aktuálního stavu přepravy různých druhů zboží a dopravního prostředku - (dopravní prostředky jsou libovolné, např. nákladní vůz, vlak, loď, lamy, saně apod.) proveďte simulaci pro zadaný počet dnů a vygenerujte do souborů následující statistiky (uložte je do vhodných souborů):

- přehled jednotlivých továren – rozpis, co kam a kdy produkovaly, kolik zboží vyprodukovaly zbytečně
- přehled jednotlivých supermarketů – každodenní rozpis skladových zásob
- celková cena přepravy $\sum_{s=1}^{S} \sum_{d=1}^{D} (c_{s,d} \sum_{z=1}^{Z} k_{s,d,z,1})$
- celková doba běhu simulace
- nemá-li úloha řešení, vypište, který den již nebylo možné supermarkety uzásobit a kolik zboží kde chybělo.

Vytvořte generátor vlastních dat. Generátor bude generovat vstupní data pomocí rovnoměrného, normálního nebo extremálního rozdělení (zvolte jedno z uvedených rozdělení, podrobnosti poskytnou cvičící na cvičeních) s vhodnými parametry. Data budou generována do souboru (nebudou přímo použita programem) o stejném formátu jako již dodané vstupní soubory. Při odevzdání přiložte jeden dataset s řešitelnou úlohou a jeden dataset, kdy se nepodaří supermarkety uzásobit.

Vytvořte dokumentační komentáře ve zdrojovém textu programu a vygenerujte programovou dokumentaci (Javadoc) a vytvořte kvalitní dále rozšiřitelný kód pro kontrolu použijte softwarový nástroj PMD (více na http://www.kiv.zcu.cz/-herout/pruzkumy/pmd/pmd.html), soubor s pravidly pmdrules.xml najdete na portálu v podmenu Samostatná práce.

V rámci dokumentace připojte zadání, popište analýzu problému, popište návrh programu (např. jednoduchý UML diagram), vytvořte uživatelskou dokumentaci, zhodnoťte celou práci, vytvořte závěr.

Analýza problému

Ze zadání vyplývá, že budeme muset vytvořit program, který bude simulovat uspokojování poptávek supermarketů při co nejnižší celkové ceně přepravy. Na první pohled se zdálo, že řešení úlohy bude probíhat pomocí reprezentace grafem a s využitím grafového algoritmu. Při bližší analýze jsme tuto možnost zavrhli a zvolíme níže popsaný alternativní přístup.

Vstupní data

Vstupní data jsou zadaná ve vstupních textových souborech. Data budeme postupně načítat a rozdělovat do potřebných matic týkajících se konkrétních továren a supermarketů. Protože soubory obsahují i slovní komentáře a prázdné řádky, samozřejmostí bude jejich ignorování.

Generátor dat

Generátor vlastních dat bude využívat normálního (Gaussovo) rozdělení. Data budou zapisována do souborů, kde forma generovaných dat bude odpovídat vzorovým data-setům přiloženým k zadání práce (úvodní informace; počty D, S, Z, T; matice počátečních zásob supermarketů, matice produkcí továren a matice poptávek supermarketů).

Simulace

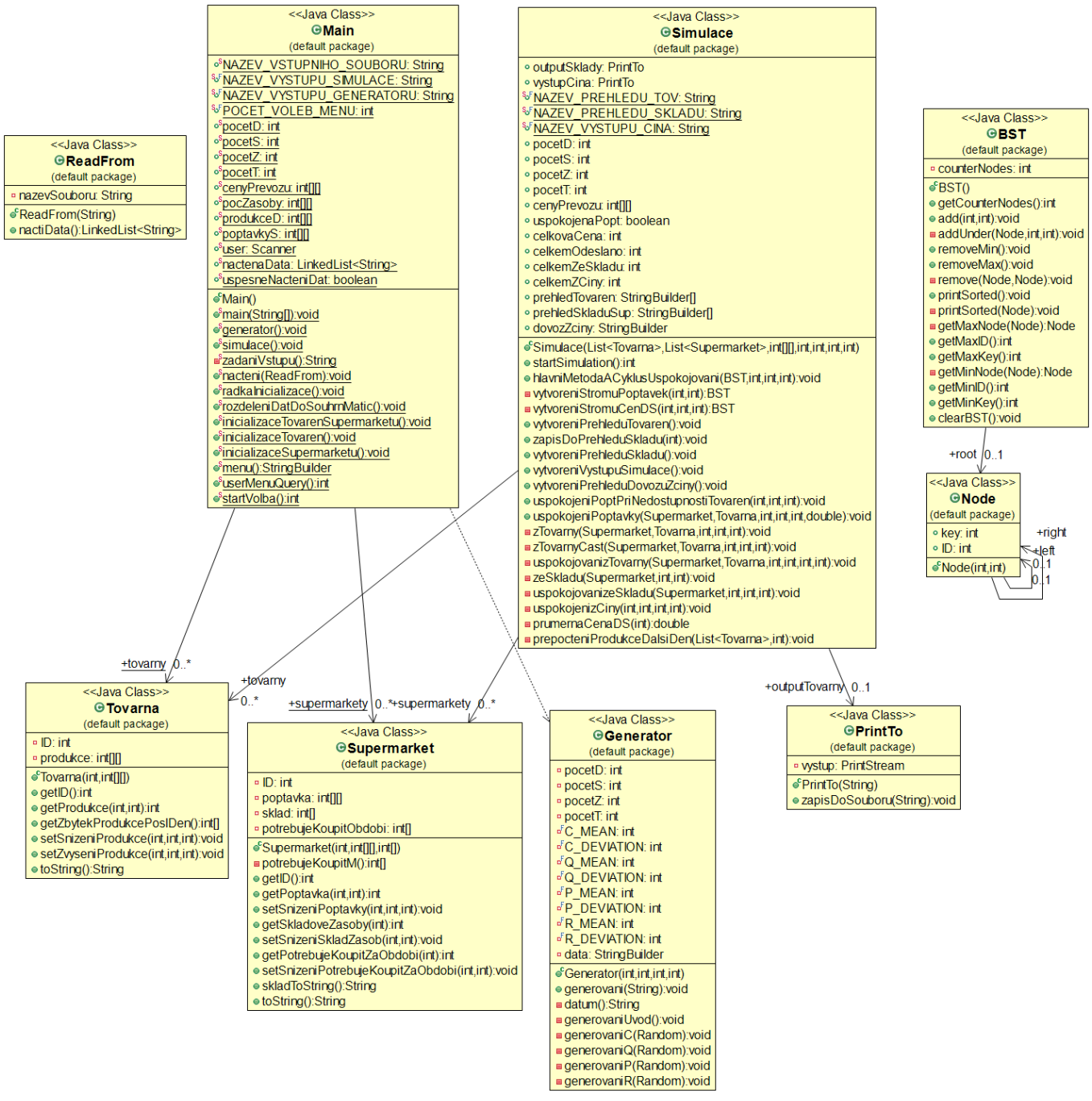
Aby byla celková cena přepravy co možná nejnižší, bude třeba začít uspokojovat největší poptávky z nejdostupnějších továren. K výběru maximálních hodnot poptávek supermarketů a minimálních hodnot cen cest budeme využívat datovou strukturu BVS. Simulace tedy bude probíhat od supermarketu s nejvyšší poptávkou až po nejnižší. Ke každému supermarketu pomocí sestaveného BVS vyhledáme nejdostupnější továrnu (s nejlevnější cenou cesty) a převezeme požadovaný počet kusů zboží. Pokud by cena cesty byla příliš drahá, budou supermarkety přednostně uspokojovat svoji poptávku ze svých počátečních zásob na skladě. V případě, že by nebylo možné supermarket uzásobit, vypíše se informace, kdy bude nutné objednat zboží z Číny.

Výstupní data

Na základě proběhlé simulace pak budeme generovat potřebné statistiky výsledků do výstupních souborů, k čemuž bude sloužit další třída pro tento zápis. Ten pak bude probíhat konkrétně do 4 textových souborů, které se vytvoří v příslušné složce.

Návrh programu

Pro bližší informace k jednotlivým třídám včetně zobrazení vazeb a závislostí přikládáme UML digram tříd (viz Obrázek 1).



Obrázek 1: UML diagram

Uživatelská dokumentace

Spuštění programu

Ke spuštění programu je potřeba mít ve složce, kde máte uložený *jar* soubor, složku se vstupními soubory (jedná se o textové soubory se vstupními daty) a složku připravenou pro soubory k zápisu dat výstupních. Jelikož se nejedná o okenní aplikaci, lze ji spustit pouze v příkazovém řádku (najdete ho zmáčknutím klávesy Windows a napsáním příkazu „cmd“), kde následně zadáte cestu k místu, kde je uložený jar soubor a nakonec příkaz: *java -jar nazevsouboru.jar*.

Ovládání programu

Po spuštění programu se v příkazovém řádku vypíše menu, které vidíte na obrázku 2. Na výběr jsou zde 3 možnosti: *Generování vstupních dat* (1), *Spustit simulaci* (2) a *EXIT* (3). Pro výběr požadavku napište jeho číslo do řádku se slovem *Volba*.

```
LUBOSUV LOGISTICKY SYSTEM
=========================
Menu
[1] Generovani vstupnich dat
[2] Spustit simulaci
[3] EXIT
-------------------------
Volba:
```

Obrázek 2: Menu

Při Zvolení možnosti „1“ – Generování vstupních dat bude program postupně vyžadovat požadovaný počet továren, počet supermarketů, počet druhů zboží a počet dnů (viz. Obrázek 3). Po zadání těchto čtyř požadavků se vygenerují nová vstupní data s těmito požadovanými počty a zapíší se do souboru *vygenerovana-data*. Vygenerovaný data-set najdete ve stejné složce, kde se nachází i Vámi spuštěný jar soubor. Tyto data pak můžete použít jako vstupní soubor pro simulaci, pokud nemáte k dispozici vlastní.

```
Pocet tovaren: 3
Pocet supermarketu: 3
Pocet druhu zbozi: 5
Pocet dnu: 2
Vygenerovana data se nachazi v souboru: vygenerovana-data.txt
=========================
```

Obrázek 3: Generování vstupních dat

Pokud zvolíte možnost „2“ - Spustit simulaci, program bude vyžadovat název vstupního souboru, se kterým chcete pracovat. Pokud je tento soubor uložen ve složce, je potřeba zadat i jméno složky (viz. Obrázek 4). Po jeho potvrzení se spustí simulace a začnou se po jednotlivých dnech a druzích zboží (Z) vypisovat informace, ze které továrny (D) do jakého supermarketu (S) se uskuteční převoz kolika kusů daného druhu zboží včetně ceny za tuto cestu. Také se vypíše celková cena přepravy za celé období, počet kusů vzatých ze skladu, počet kusů objednaných z Číny a počet kusů celkem odeslaných z továren. To je zobrazeno na obrázku 5. Zároveň se ve výstupní složce vytvoří soubory s výstupními daty.

```
Menu
[1] Generovani vstupnich dat
[2] Spustit simulaci
[3] EXIT
------------------------
Volba: 2

Zadejte nazev vstupniho souboru: vstupni-data/test_optim.txt
```

Obrázek 4: Zadání vstupního souboru

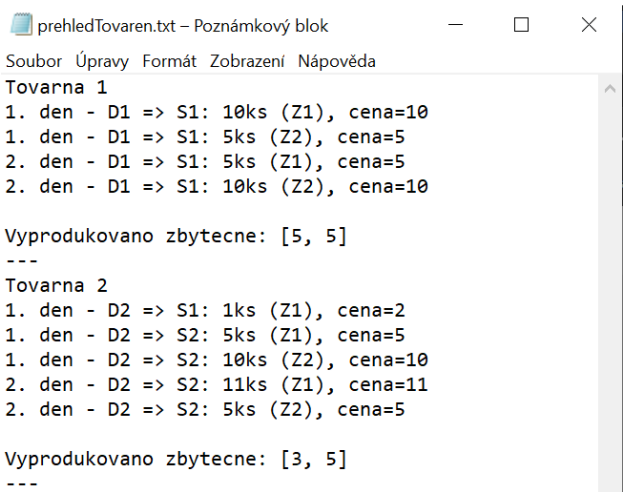
```
Spusteni simulace:
T1
Pro Z1
D1 (nakl.auto) => S1: 10ks, cena=10
sklad => S1: 1ks
D2 (nakl.auto) => S1: 1ks, cena=2
D2 (nakl.auto) => S2: 5ks, cena=5

Pro Z2
D2 (nakl.auto) => S2: 10ks, cena=10
sklad => S2: 2ks
D1 (nakl.auto) => S1: 5ks, cena=5

------------------------
T2
Pro Z1
D2 (nakl.auto) => S2: 11ks, cena=11
sklad => S2: 1ks
D1 (nakl.auto) => S1: 5ks, cena=5

Pro Z2
D1 (nakl.auto) => S1: 10ks, cena=10
sklad => S1: 2ks
D2 (nakl.auto) => S2: 5ks, cena=5

------------------------

Celkova cena prepravy za cele obdobi = 63

Celkem ze skladu: 6ks
Celkem z Ciny: 0
Celkem odeslano z tovaren: 62ks
========================
```

Obrázek 5: Průběh simulace

Popis výstupních statistik

Přehled továren

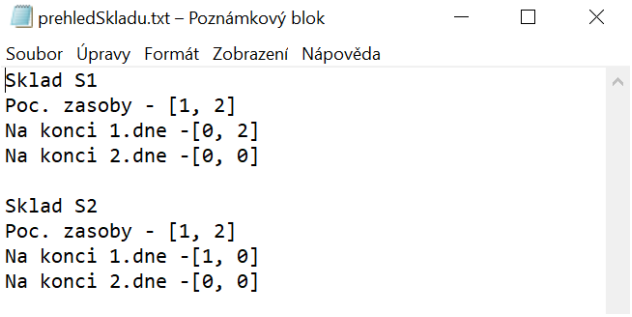
V souboru *prehledTovaren* se vygenerují přehledy pro každou továrnu ze vstupního souboru. Tyto informace zahrnují ID továrny (D), číslo dne, ve kterém se převoz uskutečnil, ID supermarketu (S), do kterého jsme zboží vezli, druh převáženého zboží (Z), jeho počet kusů a cenu tohoto převozu. Pro každou továrnu se také vypíše počet zbytečně vyrobených kusů všech druhů zboží. (viz. obrázek 6)



Obrázek 6: Přehled továren

Přehled skladů

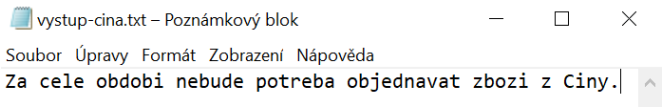
Soubor *prehledSkladu* obsahuje ID uspokojovaného supermarketu (S), informaci o počátečních zásobách pro jednotlivé druhy zboží (Z) a průběh jejich úbytků po dnech. (viz. Obrázek 7)



Obrázek 7: Přehled skladů

Čína

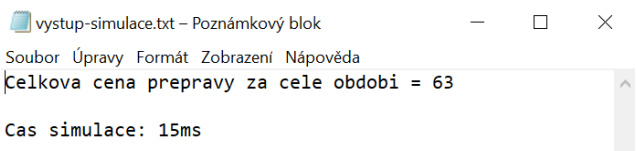
Informace o tom, zda bude (nebo nebude) nutné objednat zboží z Číny a případně v jaký den, jaký druh zboží a počet jeho kusů je uveden v souboru *vystup-cina*. (viz. Obrázek 8)



Obrázek 8: Výstup Čína

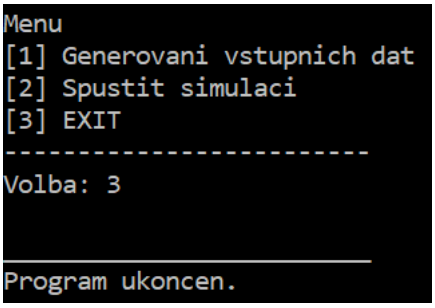
Simulace

Celkovou cenu převozu za celé období a celkovou dobu běhu simulace pak uvádí soubor vystup-simulace. (viz. Obrázek 9)



Obrázek 9: Výstup simulace

Pokud zvolíte možnost „3“ – EXIT, program se ukončí a vypíše větu „Program ukončen“. (viz. Obrázek 10)



Obrázek 10: Ukončení programu

Závěr a zhodnocení

Zadání se nám podařilo splnit v požadovaném rozsahu. Program úspěšně simuluje zásobování supermarketů. Kód je navržen i v souladu s pmd pravidly, a je tak dále rozšiřitelný. Po menších počátečních komplikacích se nám podařilo splnit i bod pro generování dat, a náš generátor úspěšně generuje a zapisuje data do souboru. Výstupem programu jsou přehledné informace o jeho průběhu vypsané v příkazovém řádku a podrobnější statistiky továren a supermarketů ve výstupních souborech.

Problémem však bylo zvolení BVS namísto pro tento problém výhodnější datové struktury Halda. Halda by umožňovala vybírání maximálních poptávek a minimálních cen v konstantním čase – O(1). Ačkoliv program funguje, jak má, algoritmus simulace pracuje ve vyšší třídě složitosti.