

Software Requirements Specification

pro
Systém pro vizualizaci Dijkstrova algoritmu

Verze 1.2
Autor: Martin Lukeš
1.12.2011

Obsah

1. Úvod.....	3
1.1 Účel dokumentu.....	3
1.2 Úvod do problematiky.....	3
2. Celkový popis.....	3
2.1 Dijkstrův algoritmus.....	3
2.1.1 Teorie grafů.....	3
2.1.2 Algoritmus.....	4
2.2 Popis aplikace.....	4
2.3 Seznam funkcí.....	5
2.4 Běhové prostředí.....	5
2.5 Dokumentace.....	6
2.6 Negativní vymezení.....	6
3. Požadavky na uživatelské rozhraní.....	7
4. Funkce aplikace.....	7
4.1 Vytvoření grafu.....	7
4.2 Výběr grafu z předpřipravených příkladů.....	7
4.3 Modifikace atributů hran a uzlů grafu.....	7
4.4 Uložení grafu do souboru a jeho opětovné nahrání.....	7
4.5 Validace atributů hran a uzlů grafu.....	8
4.6 Automatické i manuální krokování průběhu algoritmu.....	8
4.7 Textový popis kroků algoritmu.....	8
4.8 Personalizace grafu – uživatel bude mít možnost si vybrat barvy hran a uzlů.....	8
4.9 Nastavení intervalu mezi jednotlivými kroky automatického krokování.....	8
4.10 Zastavení a opětovné spuštění algoritmu od začátku.....	8
5. Instalační příručka.....	8
6. Nefunkční požadavky.....	9
6.1 Lokalizace.....	9
6.2 Požadavky na výkon.....	9
6. Přílohy.....	10
6.1 DTD pro soubory s uloženými grafy.....	10
6.2 Odkazy.....	10
6.3 Přiložené soubory.....	10

1. Úvod

1.1 Účel dokumentu

Tento dokument slouží jako specifikace požadavků na aplikaci „Systém pro vizualizaci Dijkstrova algoritmu“ (dále jen SVDA). Podrobně stanovuje rozsah projektu a funkční i nefunkční požadavky.

1.2 Úvod do problematiky

V rámci výuky předmětů na vysokých školách se studenti IT oborů seznamují se základními datovými strukturami a následně i s algoritmy, které jsou používané k efektivnímu řešení problémů v praxi.

V rámci cvičení je průběh algoritmů zobrazován na nevyhovujícím médiu - tabuli a v důsledku časového presu dochází k nedostatečnému pochopení algoritmu studenty. Míra pochopení probírané látky se pohybuje okolo 20 – 80 %.

Dalším problémem je opakovaný průchod algoritmem mimo školní prostředí. Studenti jsou odkázáni na své nedostatečné znalosti odnesené ze školních lavic, dále pak na zdroje z internetu s pestrými škálami relevance.

Tyto problémy se snaží odstranit katedra permoníků a trpaslíků z fakulty elektrotechnické vypsáním výběrového řízení za účelem vytvoření vizualizace pro malou část problému, konkrétně Dijkstrův algoritmus. SVDA je projektem studentů fakulty elektrotechnické účastnících se onoho výběrového řízení.

2. Celkový popis

2.1 Dijkstrův algoritmus

2.1.1 Teorie grafů

V této sekci následuje množina termínů potřebná k definování a pochopení algoritmu.

Graf	je uspořádaná dvojice množiny uzlů a množiny hran. V praxi značeno (V, E) . Grafy dělí na orientované a neorientované. Neorientované grafy obsahují jen neorientované hrany, orientované grafy obsahují jen orientované hrany.
Uzel	je jednoznačně identifikovaný bod. Množina N uzlů bývá v praxi značena posloupností přirozených čísel $1 \dots N$. Jednoznačnost si můžeme přeložit tak, že v množině uzlů neexistují uzly se stejným číslem. Bod je nejjednodušším grafickým znázorněním uzlu.
Hrana	je jednoznačně definovanou spojnici mezi dvěma uzly. Množina N uzlů bývá v praxi značena posloupností písmen latinské abecedy, přičemž v případě vyššího počtu uzlů

se nejedná o znaky, nýbrž o řetězce vytvořené z těchto znaků. Hrana v množině je jednoznačná, pokud neexistuje jiná hrana se stejným písmenem (řetězcem). Hrana definuje vztah mezi dvěma uzly grafu. Abstrakce vztahu uzlů je dvojice uzlů. Pro neznalého čtenáře se může zdát, že hrana jasně identifikuje hranu, nicméně není tomu tak vždy. V množině hran mohou existovat různé hrany spojující stejné uzly s různým řetězcem. Takovýto graf potom obsahuje tzv. multihrany.

Neorientovaná hrana definuje vztah mezi dvěma uzly jako neuspořádanou dvojici například (a, b) nebo (b, a) obě tyto hrany spojují dva stejné uzly a obecně u nich nedokážeme říci, zda jsou různé – v grafu neobsahujícím multihrany jde o jednu a tu samou hranu, v obecném grafu o této skutečnosti nedokážeme rozhodnout. Orientovaná hrana je reprezentovaná uspořádanou dvojicí. U orientované hrany rozeznáváme pořadí uzlů, první z uzlů nazýváme počátek, druhý potom konec hrany.

Takže například orientovaná hrana (a, b) spojuje stejné uzly jako hrana (b, a), ale jistojistě tyto hrany nejsou shodné. V teorii grafů je možné každou hranu ohodnotit, potom přidáme do grafu k definici dvojic i hodnotící funkci, která ke každé hraně přiřadí právě jednu hodnotu celého čísla.

Cesta je posloupnost uzlů a hran, přičemž tato posloupnost začíná uzlem, končí uzlem, žádné dva uzly po sobě bezprostředně nenásledují a žádné dvě hrany po sobě bezprostředně nenásledují.

Uzel je obsažen v dvojici každé následující hrany a uzel následující hranu je také obsažen v dvojici hrany, takže hrana určuje vazbu mezi těmito hranami. V následujícím textu je vynechán identifikátor hrany, hrana je potom znázorněna jako dvojice uzlů. Cestou je například posloupnost 1, (1, 2), 2, (2, 3), 3, (3, 4), 4. Tato cesta může být orientovaná nebo neorientovaná. Cesta může existovat v orientovaném či neorientovaném grafu. Posloupnost 1 (2, 1) 2, (2, 3), 3 je potom jistojistě neorientovaná.

Váha cesty je potom suma všech vah hran cesty. Nejkratší cesta je potom taková, že neexistuje cesta s menší vahou nežli je váha nejkratší cesty.

2.1.2 Algoritmus

Dijkstrův algoritmus patří mezi grafové algoritmy a slouží k nalezení minimální cesty v grafu mezi jedním uzlem nazývaným počátek a zbylými uzly. Tento algoritmus ohodnocuje každý uzel reálným číslem značícím vzdálenost uzlu od počátku (délku nejkratší cesty). Algoritmus funguje korektně v orientovaných i neorientovaných grafech. Algoritmus funguje taktéž korektně v grafech obsahujících multihrany. Nicméně algoritmus nemusí dávat korektní výsledky v grafu s záporně ohodnocenými hranami.

2.2 Popis aplikace

Aplikace bude umožňovat zobrazení průběhu Dijkstrova algoritmu popsaného v skriptech Doc. RNDr. Josef Koláře, CSc uveřejněných na internetových stránkách [1] (viz sekce odkazy) přiložených v přílohách v příloze [2] v sekci 6.2 na stranách 116 - 118 .

Kromě samotné vizualizace na příkladových grafech bude aplikace umožňovat uživatelům

vytvořit konečný graf libovolné velikosti s libovolným nezáporným ohodnocením hran. Aplikace bude také umožňovat uložení grafů a opětovné nahrání uložených grafů ze souboru.

Aplikace bude dále umožňovat opakovaný průchod algoritmem. Aplikace bude umožňovat krokování uživatelem či automatické krokování. Aplikace bude umožňovat nastavení časového intervalu mezi jednotlivými kroky automatického krokování. Aplikace bude umožňovat nastavení personalizace barev hran a uzlů.

2.3 Seznam funkcí

- Vytvoření grafu
- Výběr grafu z před-připravených příkladů
- Modifikace atributů hran a uzlů grafu
- Uložení grafu do souboru a jeho opětovné nahrání
- Validace atributů hran a uzlů grafu
- Automatické i manuální krokování průběhu algoritmu
- Textový popis kroků algoritmu
- Personalizace grafu – uživatel bude mít možnost si vybrat barvy hran a uzlů
- Nastavení intervalu mezi jednotlivými kroky automatického krokování
- Zastavení a opětovné spuštění algoritmu od začátku
- Aplikace neumožňuje ukládat stav grafu v průběhu algoritmu, uložené grafy ve formátu .xml nebudou obsahovat informaci o ohodnocení uzlů
- Aplikace nebude umožňovat vracet se v průběhu algoritmu o žádný krok zpět. Výjimkou je, že uživatel bude moci spustit algoritmus od začátku jeho běhu

2.4 Běhové prostředí

Projekt bude realizován ve formě desktopové aplikace. Objem instalačních souborů aplikace nepřesáhne velikost kompaktního disku CD (700 MB), objem vlastní aplikace potom 2 GB. Instalace ani běh aplikace nevyžaduje připojení k internetu. Minimální požadovaná konfigurace aplikace je Pentium Dual Core 1,2 Ghz 1 GB RAM. Optimální konfigurace Pentium Dual Core 2 Ghz 2 GB RAM. Aplikace poběží v uživatelských prostředích Microsoft Windows XP, Microsoft Windows Vista, Microsoft Windows 7, Linux Ubuntu 11.10.

2.5 Dokumentace

Součástí projektu je následující dokumentace:

- Uživatelská dokumentace – Bude integrována přímo do uživatelského rozhraní formou kontextových nápověd. Kontextové menu bude přístupné v panelu Help.
- Instalační a konfigurační příručka – Dokument ve formátu PDF popisující instalaci a konfiguraci krok za krokem. Součástí dokumentu bude také reference všech nastavitelných vlastností aplikace. Tento dokument bude pouze v anglickém jazyce.
- Reference zdrojového kódu – Reference zdrojového kódu v HTML či PDF generovaná dokumentačním nástrojem Doxygen. Tento dokument bude pouze v anglickém jazyce.
- Programátorská reference – PDF dokument popisující architekturu celé aplikace a implementační detaily (členění kódu). Tento dokument bude pouze v anglickém jazyce.

2.6 Negativní vymezení

- Aplikace nebude realizovaná jako webovou aplikaci, nebude spustitelná z internetového prohlížeče
- Aplikace nebude umožňovat ukládání obrázků grafů
- Aplikace nebude validovat graf z hlediska existence záporných cyklů v grafu
- Aplikace nebude umožňovat spuštění algoritmu nad grafem s záporně ohodnocenými hranami
- Aplikace nebude umožňovat ukládání nebo načítání grafů z jiného formátu nežli .xml validního vůči Document Type Definition v příloze [2]
- U operačních systémů nevyjmenovaných v seznamu v bodě 2.4 není garantovaná správná funkce aplikace.

2.7 Předpokládaná další rozšíření

Aplikace je rozšiřitelná v budoucnosti zejména, co se týče transformace na webovou aplikaci. Dalším možným rozšířením aplikace je generování obrázků grafů, automatické rozprostření grafu po ploše editoru. Větším, ale očekávaným rozšířením aplikace je přidání dalších grafových algoritmů do seznamu vizualizovaných, ať už se jedná o grafové algoritmy pro vyhledávání nejkratší cesty, jiných grafových algoritmů či algoritmů pro řazení.

3. Požadavky na uživatelské rozhraní

- Algoritmus bude vizualizován v centrální části okna aplikace.
- Pod lištami nabídek budou tlačítka a pole pro výběr pozice, barvy uzlů.
- Na pravém okraji od centrální oblasti budou nástroje pro vytvoření hrany, uzlu, posunutí, výběr a vytvoření barvy, uložení grafu a nahrání grafu.
- Barevné schéma aplikace bude přizpůsobené uživatelskému look & feel (look & feel operačního systému).

Rozmístění komponent bude pro pohodlnou práci vyžadovat, aby okno prohlížeče mělo velikost nejméně 800x600 obrazkových bodů.

4. Funkce aplikace

4.1 Vytvoření grafu

Uživatel bude mít možnost nadefinovat si a umístit uzly reprezentované body na obrazovce, ať už pomocí pozice či myši. Dále bude mít možnost nadefinovat jim jejich labely v případě, že nedochází k kolizi s labely jiných uzlů.

4.2 Výběr grafu z předpřipravených příkladů

Uživatel bude mít možnost načíst si nejméně 10 předpřipravených grafů a na ně aplikovat algoritmus. Tyto grafy bude možné modifikovat a uložit, nicméně původní ukázkové příklady v aplikaci nezmizí.

4.3 Modifikace atributů hran a uzlů grafu

Uživatel bude mít možnost při vytváření grafu změnit pozici uzlů, zlomení hran (čar reprezentujících hrany). Smazání hran a uzlů grafu. Kromě těchto přirozených atributů bude uživatel moci nastavit nekolizní (unikátní v rámci grafu) labely uzlů a hran.

4.4 Uložení grafu do souboru a jeho opětovné nahrání

Uživatel bude mít možnost perzistentního uložení grafu, na který je aplikován algoritmus. Graf se uloží do souboru s názvem definovaným uživatelem s příponou .xml.

4.5 Validace atributů hran a uzlů grafu

Aplikace bude validovat jednoznačnost uzlů (jejich labelů) a hran (jejich labelů). Dále bude aplikace validovat neexistenci záporných hran grafu.

4.6 Automatické i manuální krokování průběhu algoritmu

Uživatel bude mít možnost v libovolném okamžiku zastavit běh algoritmu a dále algoritmus krokovat nebo naopak spustit automatické krokování.

4.7 Textový popis kroků algoritmu

Algoritmus bude stručně a přehledně upozorňovat uživatele o krocích, které provádí, tento popis bude inspirován popisem algoritmu ve skriptech viz [1].

4.8 Personalizace grafu – uživatel bude mít možnost si vybrat barvy hran a uzlů

Uživatel bude mít možnost vybrat si jednu z 10 nabízených barev pro relaxované hrany, vybrané hrany, nevybrané hrany, uzavřené uzly. Krom toho bude mít uživatel možnost vytvořit si barvu definovanou pomocí složek RGB a uložit si ji místo libovolné barvy z 10 defaultních.

4.9 Nastavení intervalu mezi jednotlivými kroky automatického krokování

Uživatel bude mít možnost nastavit si interval čekání mezi jednotlivými kroky algoritmu. Tento interval nezahrnuje výpočet algoritmu, takže je nerelevantní pro slabší než optimální konfigurace a grafy s více než 100 Uzly a 10000 hran.

4.10 Zastavení a opětovné spuštění algoritmu od začátku

Uživatel bude mít možnost zastavit a znovu spustit algoritmus od začátku jeho běhu. Tato funkcionality nemá nic společného s zpětným krokováním algoritmu.

5. Instalační příručka

Aplikace se bude instalovat přes instalačního blizard. Správný běh aplikace je zaručen na zařízení s minimální a lepší konfigurací. V případě instalace na horší zařízení bude uživatel upozorněn na nedostatečnou konfiguraci a tudíž na nemožnost garance správné funkčnosti aplikace.

6. Nefunkční požadavky

6.1 Lokalizace

Pro aplikaci budou dostupné česká a anglická lokalizace aplikačního rozhraní a kontextových menu.

6.2 Požadavky na výkon

Minimální požadovaná konfigurace aplikace je Pentium Dual Core 1,2 Ghz 1 GB RAM. Optimální konfigurace Pentium Dual Core 2 Ghz 2 GB RAM. Aplikace musí zvládat vizualizaci středně velkých až velkých grafů, tj grafů s maximálně 1000 uzly a 10000 hranami s minimální nastavenou délkou automatického kroku a intervalem mezi jednotlivými kroky algoritmu 1s. Délka intervalu je měřena pro optimální konfiguraci.

6. Přílohy

6.1 DTD pro soubory s uloženými grafy.

```
<?xml version='1.0' encoding='UTF-8'?>
  <!ELEMENT graph (edges|nodes)*>
    <!ELEMENT nodes (node)*>
    <!ELEMENT node EMPTY>
  <!ATTLIST node   label CDATA #IMPLIED >
  <!ELEMENT edges (directedEdge|edge)*>
    <!ELEMENT edge EMPTY>
    <!ATTLIST edge
      weight CDATA #IMPLIED
      endpoint CDATA #IMPLIED
      startPoint CDATA #IMPLIED
      label CDATA #IMPLIED >
  <!ELEMENT directedEdge EMPTY>
    <!ATTLIST directedEdge
      weight CDATA #IMPLIED
      endpoint CDATA #IMPLIED
      startPoint CDATA #IMPLIED
      label CDATA #IMPLIED >
```

6.2 Odkazy

[1] <http://service.felk.cvut.cz/courses/36TI/>

6.3 Přiložené soubory

[2] Skripta doc. Jiřího Koláře kap06.pdf