Zadání projektu z předmětu Grafové algoritmy (GAL)

Obecný úvod k projektům:

Rešeršní úlohy jsou typicky pro jednoho studenta (individuální). V případě náročnější implementační úlohy jsou projekty týmové po dvou studentech (ve výjimečných případech lze řešit projekt jednotlivě, ale očekává se vypracování projektu téměř v celé jeho šíři, tj. nedojde k odpovídajícímu snížení nároků jen z toho důvodu, že je student dobrovolně sám). **Varianta projektu se registruje ve WIS** na začátku semestru. Nejprve se registruje první člen týmu (kapitán) a následně po navýšení kapacit u týmových variant se doregistruje jeho kolega. Pozice kapitána (vedoucího) je většinou pouze formální, avšak v případě neshod v týmu rozhoduje o řešení vedoucí po domluvě s cvičícím. Veškerá **komunikace elektronickou poštou** se cvičícím musí obsahovat **jako adresáta kopie i druhého člena týmu**! Data budou rozepsána u popisu termínu *Projekt* ve WIS. V případě, že máte zájem o vlastní zadání, tak pošlete email cvičícímu nebo přednášejícímu s návrhem, o co by se mělo jednat, a seznamem studijní literatury.

Odevzdávání:

Projekt, zabalený v zip nebo tar+gzip a pojmenovaný *login*.(zip|tar.gz), kde *login* je vaše fakultní přihlašovací jméno, se odevzdává do informačního systému do zvolené varianty. Shodnou verzi (až na pojmenování souboru) odevzdávají oba členové týmu, nikoliv jen vedoucí! V případě rešeršní úlohy, kde je výsledkem pouze jeden dokument PDF se odevzdává pouze jediný soubor pojmenovaný gal20-*login*.pdf. **Termín odevzdání projektu je dle IS FIT (termín Projekt)**. Po tomto datu nelze projekt odevzdat. Neodevzdaný projekt je automaticky za 0 bodů. Taktéž nelze projekt odevzdat e-mailem -- je třeba jej nahrát do informačního systému. Pokud projekt přesáhne velikost 1,5 MB, domluvte si jiný způsob odevzdání se cvičícím ještě před odevzdáním projektu. Projekt je nutné obhájit, a to buď na studentské konferenci LTA (http://www.fit.vutbr.cz/~meduna/work/lta) pořádané na fakultě (prezentace v angličtině), která se bude konat koncem semestru (termín bude včas upřesněn), nebo na vypsaných termínech obhajob (během (před)poslední přednášky nebo jindy; prezentace v češtině). Za vynikající prezentaci mohou být uděleny bonusové body.

Implementace:

U implementace se předpokládá, že bude přeložitelná a spustitelná na serveru Merlin. Podle toho tedy volte programovací prostředí (programovací jazyk, překladač, knihovny, apod.). U demonstračních projektů si lze (po domluvě s cvičícím) zvolit i jiná prostředí. V případě mimořádně kvalitního vypracování projektu lze udělit i bonusové body.

Dokumentace:

Ke každému projektu musí být vypracována dokumentace, jejíž rozsah a obsah bude závislý na zvolené variantě projektu. Nicméně, v každé dokumentaci musí být uvedeny všechny zdroje, knihovny apod., ze kterých jste při vypracování čerpali a které jste využili. Neuvedený zdroj se bere jako podvod (vydáváte cizí řešení za své). Dále každá dokumentace musí obsahovat návrh programu, popis jeho ovládání. V případě, že se v zadání projektu vyžadují experimenty, tak odevzdaný projekt musí obsahovat zdrojová data pro tyto experimenty (u velkých datových zdrojů uveďte v dokumentaci jejich URL) a dokumentace musí obsahovat popis experimentů a závěr, který z nich vyplývá. Na dokumentaci a experimenty bude při hodnocení kladen velký důraz.

Konzultace a dotazy:

Jednoduché dotazy lze řešit emailem. V ostatních případech si lze domluvit konzultaci (viz konzultační hodiny u cvičícího nebo po přednášce u přednášejícího).

Konzultace (cvičící):

- Martin Tomko, místnost C220, email: itomko@fit.vut.cz
- Zbyněk Křivka, místnost C229, email: krivka@fit.vut.cz

Následuje popis typů zadání (písmeny číslované nadpisy). Varianty, ze kterých lze vybírat při registraci projektu do GAL, jsou potom k dispozici u variantního termínu „Projekt“.

(A) Zadání pro demonstrační úlohy

Vytvořte demonstrační aplikaci, která názorně vysvětluje použití a práci vybraného algoritmu. Při vytváření dbejte na příjemné uživatelské rozhraní, intuitivnost ovládání a možnost využití pro didaktické účely (důležitá část hodnocení). **Uživatelské rozhraní před implementací konzultujte se zodpovědným cvičícím**. Aplikace bude umožňovat načítání a ukládání zadaných grafů, základní vizuální editaci grafů a následně demonstraci algoritmu (podpora konfigurovatelných klávesových zkratek výhodou pro ovládání prezentérem). Pro vstupní a výstupní formát grafu uvažujte notaci GraphML nebo GML, po dohodě se cvičícím lze použít i jiný formát. Dokumentace bude doplněna i o krátký úvodní popis možností aplikace, seznam minimálních požadavků (architektura, operační systém, podpůrný software) a tutoriál použití aplikace (uživatelská příručka). Použití aplikace demonstrujte na ukázkových příkladech, které budou též součástí odevzdaného archivu. Prvotní doporučenou literaturu rozšiřte i o další zdroje jako odborné články autorů algoritmů, jiné existující implementace a další. Dokumentace bude též obsahovat popis implementovaného algoritmu v notaci blízké přednáškám.

(B) Rešerše: Zpracování vybraných témat

Podrobně nastudujte vybraný problém a individuálně vypracujte odbornou rešerši v češtině nebo slovenštině (v případě diskutabilních překladů uveďte v závorce i pojem v originálním znění). Při popisu se zaměřte na hlavní principy, vysvětlete problematiku za použití vašich vlastních vzorových příkladů, ukázek použití v praxi apod. Neopakujte pojmy z přednášek, ale využívejte terminologii i notaci prezentovanou na přednáškách. Text by měl být vhodný pro absolventy předmětu GAL. V některých zadáních je vyžadováno i řešení vybraných příkladů. Při zpracování postupujte dle obvyklých typografických zásad. Výsledek práce bude v závěru semestru studentem osobně prezentován: Buď česky na obhajobách GAL, nebo v angličtině na studentské konferenci LTA (Language Theory with Applications).

Rozsah práce je stanoven na cca 20 normostran (formát A4; kompatibilní s černobílým tiskem). Do rozsahu se započítávají vhodně zvolené obrázky, tabulky a diagramy. Uvedená literatura je pouze pro uvedení do problematiky, další zdroje musí student vyhledat sám. Připomínáme, že fakulta má přístup do různých databází článků. Pro možnost stažení plného textu článku je často nutné přistupovat na stránky databáze ze sítě fakulty. V případě, že se Vám nedaří získat relevantní článek, kontaktujte cvičícího/přednášejícího nebo knihovnu FIT. Kromě samotného obsahu a jeho návaznosti na přednášenou látku bude hodnocen i pravopis, stylistika, didaktičnost výkladu a vlastní přínos (příklady apod.).

(C) Řešení paralelizace a měření časové a prostorové složitosti

Implementujte jak sekvenční, tak paralelní verzi zvoleného algoritmu. Experimentálně vyhodnoťte časové a prostorové složitosti algoritmů nad dostatečně velkými grafy s využitím dostatečného počtu testů. Návrh testů zdůvodněte v dokumentaci. Porovnejte

výsledky sekvenčního a paralelního řešení z hlediska jejich časové a případně i prostorové složitosti. Experimenty provádějte na vícejádrovém/víceprocesorovém systému a při vyhodnocení vezměte v úvahu také množství použitých jader/procesorů. Nakonec experimentální výsledky (alespoň pro sekvenční verzi) porovnejte s teoretickými a zdůvodněte případný nesoulad.

I přes relativní volnost ohledně použitého implementačního jazyka silně doporučujeme využít kompilovaný jazyk (např. C/C++ nebo i Haskell), který nabízí dobře optimalizovaný překladač (GCC, Visual Studio, Intel, LLVM, GHC, ...).

V případě zájmu o studentský přístup k superpočítači Salomon se hlaste co nejdříve na začátku semestru přednášejícímu. Modul superpočítače umožňuje plnou dedikaci hardware pro danou softwarovou úlohu s využitím většího počtu jader a kapacity paměti.

(D) Porovnání dvou algoritmů

Nastudujte, implementujte a porovnejte dva Vámi vybrané algoritmy řešící zadaný problém, z nichž jeden může být přednášen (specifikováno v konkrétní variantě zadání) a minimálně jeden nesmí být přednášen (v případě nejasností konzultujte se cvičícím). Experimentálně vyhodnoťte časové a prostorové složitosti algoritmů nad dostatečně velkými problémy s využitím dostatečného počtu testů. Návrh testů zdůvodněte v dokumentaci. Proveďte porovnání časové složitosti algoritmů (teoretické i experimentálně zjištěné), případně prostorové složitosti. Dále srovnejte vhodnost těchto algoritmů pro různé typy grafů (řídké/husté grafy, ...). Dosažené výsledky zdůvodněte. Implementace a porovnání více než dvou algoritmů může být hodnoceno bonusovými body. Výstupem projektu tohoto typu bude konzolová aplikace a zdokumentované závěry z experimentů.

(E) Speciální zadání

Zadání je upřesněno u konkrétní varianty. V případě nejasností je třeba konzultovat se cvičícím.

Reference:

[B08] J. A. Bondy, U. S. R. Murty: *Graph Theory*, Springer, 2008.

[B09] R. Burkard et al.: *Assignment Problems,* SIAM, 2009

[C02] T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to Algorithms*, McGraw-Hill, 2002.

[C09] T.H. Cormen, C.E. Leiserson, R.L. Rivest: *Introduction to Algorithms*, 3rd Edition, McGraw-Hill, 2009.

[D02] J. Demel: *Grafy a jejich aplikace*, nakladatelství Academia, Praha 2002.

[E92] J. R. Evans, E. Minieka: *Optimization Algorithms for Networks and Graphs*, Second Edition, Marcel Dekker, New York, 1992.

[E12] S. Even: *Graph Algorithms*, 2nd Edition, Cambridge University Press, 2012.

[G85] A. Gibbons: *Algorithmic Graph Theory*, Cambridge University Press, 1985.

[H74] J. Hopcroft, R. E. Tarjan, Efficient planarity testing. J. Assoc. Comput. Mach. **21**(4), s. 549-568, 1974.

[K06] J. Kleinberg, É. Tardos: *Algorithm Design*, Pearson/Addison-Wesley, 2006.

[K09] M. Krauter: Nejkratší cesty v grafu, diplomová práce, FIT VUT v Brně, Brno, 2009.