

Zadání projektu z předmětu Grafové algoritmy (GAL)

Obecný úvod k projektům:

Rešerní úlohy jsou typicky pro jednoho studenta (individuální). V případě implementační úlohy jsou projekty týmové po dvou lidech (ve výjimečných případech lze řešit projekt jednotlivě, ale očekává se vypracování projektu v celé jeho šíři, tj. nedojde ke snížení nároků jen z toho důvodu, že je student sám). **Varianta projektu se registruje ve WIS** na začátku semestru. Nejprve se přihlásí pouze vedoucí (kapacita všech variant zadání bude 1). Poté bude u implementačních zadání zvýšena kapacita na 2 a umožněno registrování i druhému členovi týmu. Pozice 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!** Datумы budou rozepsány 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 se odevzdává pouze jediný soubor pojmenovaný *login.pdf*. **Termín odevzdání projektu je 14. 12. 2014** do 23:59:59 včetně. 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 2 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 vypsanych termínech obhajob (během poslední přednášky nebo jindy; prezentace v češtině). Za vynikající prezentaci mohou být uděleny bonusové body.

Implementace:

U implementačních projektů se předpokládá, že budou přeložitelné a spustitelné 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).

Konzultace (cvičící): Zbyněk Křivka, místnost C229, email: krivka@fit.vutbr.cz

Následuje popis typů zadání (písmeny číslované nadpisy) a následně jejich variant (bez číslování), ze kterých lze vybírat při registraci projektu do GAL.

(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. Aplikace bude umožňovat načítání a ukládání zadáných grafů, základní vizuální editaci grafů a následně demonstraci algoritmu. Pro vstupní a výstupní formát grafu uvažujte notaci GraphML, 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 a další.

Algoritmus Out of Kilter (toky v síti)

Viz Zadání pro demonstrační úlohy.

Literatura:

- Jiří Demel: *Grafy a jejich aplikace*, nakladatelství Academia, Praha 2002, ISBN 80-220-0990-6

Egerváryho algoritmus (párování v bipartitních grafech)

Viz Zadání pro demonstrační úlohy.

Literatura:

- J. A. Bondy, U. S. R. Murty, *Graph Theory*, Springer, 2008. Chapter 16.

Fleuryho algoritmus (Eulerovské tahy)

Viz Zadání pro demonstrační úlohy.

Literatura:

- J. A. Bondy, U. S. R. Murty, *Graph Theory*, Springer, 2008. Chapter 3.

Barvení hran - Problém minimalizace

Po dohodě s cvičícím vyberte algoritmus pro barvení hran vhodný pro řešení problému minimalizace. Dále viz Zadání pro demonstrační úlohy.

Literatura:

- Krauter Michal: *Nejkratší cesty v grafu*, diplomová práce, FIT VUT v Brně, Brno, 2009.

Fibonacci heap

Prostudujte Fibonacciho haldy a po dohodě se cvičícím vyberte několik operací nad těmito haldami a vytvořte pro ně didaktickou demonstrační aplikaci. Dále viz Zadání pro

demonstrační úlohy.

Literatura:

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

Disjoint Sets

Prostudujte problematiku *disjoint-set datových struktur* a demonstруйте operace *make-set*, *union* a *find-set* nad těmito datovými strukturami. Demonstруйте i různé heuristiky. Dále viz Zadání pro demonstrační úlohy.

Literatura:

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

Push-relabel algoritmus

Prostudujte problematiku operací *push* a *relabel* a jejich činnost demonstруйте vytvořením didaktické aplikace. Dále viz Zadání pro demonstrační úlohy.

Literatura:

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

Relabel-to-front algoritmus

Prostudujte problematiku *relabel-to-front* algoritmu, vytváření „seznamu sousedů“ a operace *discharge* a činnost algoritmu demonstруйте vytvořením didaktické aplikace. Dále viz Zadání pro demonstrační úlohy.

Literatura:

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

Hledání cyklů v orientovaném grafu

Prostudujte algoritmus navržený D. B. Johnsonem. Dále viz Zadání pro demonstrační úlohy.

Literatura:

- D. B. Johnson, *Finding All Elementary Circuits of a Directed Graph*, SIAM J. Comput., 1975.

Edmonds-Karpův algoritmus (vyhledání max. toku)

Viz Zadání pro demonstrační úlohy.

Literatura:

- J. A. Bondy, U. S. R. Murty, *Graph Theory*, Springer, 2008.
- T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 2002.

Dinicův algoritmus (vyhledání max. toku)

Viz Zadání pro demonstrační úlohy. Téma vyžaduje samostudium literatury:

http://en.wikipedia.org/wiki/Dinic%27s_algorithm#References

Edmondsův párovací algoritmus (vyhledání max. párování)

Viz Zadání pro demonstrační úlohy. Téma vyžaduje samostudium literatury:
http://en.wikipedia.org/wiki/Edmonds%27s_matching_algorithm#References

Kosarajův algoritmus (vyhledání silně s. komponent)

Viz Zadání pro demonstrační úlohy.

Gabowův algoritmus (vyhledání silně s. komponent)

Viz Zadání pro demonstrační úlohy. Téma vyžaduje samostudium literatury:
http://en.wikipedia.org/wiki/Gabow%27s_algorithm#References

Chu-Liu/Edmondsův algoritmus (vyhledání optimálního větvení)

Viz Zadání pro demonstrační úlohy. Téma vyžaduje samostudium literatury:
http://en.wikipedia.org/wiki/Chu-Liu/Edmonds_algorithm#References

Tarjanův algoritmus (vyhledání optimálního větvení)

Viz Zadání pro demonstrační úlohy. Téma vyžaduje samostudium literatury:
<http://www.scribd.com/doc/6853671/Tarjan-Finding-optimum-branchings>

Hopcroft-Karpův algoritmus (hledání maximálního párování)

Téma vyžaduje samostudium literatury: http://en.wikipedia.org/wiki/Hopcroft%E2%80%93Karp_algorithm#References

Literatura:

- Rainer Burkard et al.: *Assignment Problems*, Section 3.3. SIAM, 2009

Testování planarity grafu

Viz Zadání pro demonstrační úlohy.

Literatura:

- A. Gibbons: *Algorithmic Graph Theory*, Section 3.4. Cambridge University Press, 1985.
- J. Hopcroft, R. Tarjan, Efficient planarity testing. J. Assoc. Comput. Mach. **21**, p. 549-568, 1974.

Hledání planární reprezentace grafu

Viz Zadání pro demonstrační úlohy.

Literatura:

- J. A. Bondy, U. S. R. Murty, Graph Theory, Springer, 2008. Chapter 10.

Optimální větvení

Viz Zadání pro demonstrační úlohy. Anglicky „Optimum Branching“.

Literatura:

- A. Gibbons: *Algorithmic Graph Theory*, Section 2.1.1. Cambridge University Press, 1985

Hledání a počítání Eulerovských tahů

Viz Zadání pro demonstrační úlohy. Anglicky „Finding and Counting Eulerian circuits“.

Literatura:

- A. Gibbons: *Algorithmic Graph Theory*, Chapter 6. Cambridge University Press, 1985.

Problém rozložených průhledných obdélníkových fólií

Navrhněte a demonstруйте korektnost algoritmu, který řeší problém identifikace průhledných fólií. Mějme na sobě položené průhledné obdélníkové fólie (ortogonálně), které na sobě mají každá uvedené unikátní přirozené číslo. Jak určíme, která je která? Resp. při překreslení n průhledných obdélníků přes sebe, kdy tyto obdélníky obsahují identifikační číslo, je třeba určit, kterému obdélníku je přiřazeno které číslo. Více viz Zadání pro demonstrační úlohy a odkaz v literatuře.

Literatura:

- M. Kačer, Toky/párování/konektivita, Snímek 40 [prezentace], 2011. K dispozici na http://www.fit.vutbr.cz/study/courses/GAL/private/materials/sorting_slides.pdf

Konstrukce chromatického polynomu

Demonstруйте obě probírané metody (postup k úplnému i izolovanému grafu) konstrukce chromatického polynomu pro zadaný neorientovaný graf. Demonstrace bude v případě zájmu uživatele podporovat i zkratky (např. polynomy pro stromy apod.). Aplikace by měla podporovat i úpravu na normální tvar $c_n k^n + c_{n-1} k^{n-1} + \dots c_1 k + c_0$.

Literatura:

- A. Gibbons: *Algorithmic Graph Theory*, Chapter 7. Cambridge University Press, 1985.

(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 originálních vzorových příkladů, ukázek použití v praxi apod. 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 2014 (Language Theory with Applications).

Rozsah práce je stanoven na 10-15 stran (formát A4). 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.

Alokace registrů pomocí barvení

Prostudujte barvení grafů se zaměřením na využití barvení grafů při alokaci registrů v překladačích. Dále viz zadání Rešerše.

Literatura:

- Briggs, P., Cooper, K.D., Kennedy, K., Torczon, L.: *Coloring heuristics for register allocation*, PLDI '89 Proceedings of the ACM SIGPLAN 1989 Conference on Programming language design and implementation, pages 275 – 284. [<http://portal.acm.org/citation.cfm?id=74843>]
- Briggs, P., Cooper, K.D., Torczon, L.: *Improvements to graph coloring register allocation*, ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 16 Issue 3, May 1994, pages 428 – 455. [<http://portal.acm.org/citation.cfm?id=177492.177575>]
- *Tailoring Graph-coloring Register Allocation For Runtime Compilation* [online], [cit. 27. 9. 2014], <http://llvm.org/pubs/2006-04-04-CGO-GraphColoring.html>

Goldberg-Radzik algoritmus

Práce se zaměří na popis algoritmu hledání nejkratších cest dle autorů Goldberg, Radzik. Dále viz zadání Rešerše.

Literatura:

- Krauter Michal: *Nejkratší cesty v grafu*, diplomová práce, FIT VUT v Brně, Brno, 2009.

Dantzigův algoritmus – řešení úlohy pomocí grafů

Zpracujte řešení dopravního problému za použití teorie grafů. Dále viz zadání Rešerše.

Literatura:

- Krauter Michal: *Nejkratší cesty v grafu*, diplomová práce, FIT VUT v Brně, Brno, 2009.
- Jiří Demel: *Grafy a jejich aplikace*, nakladatelství Academia, Praha 2002, ISBN 80-220-0990-6

Max-Flow Min-Cut teorém

Vyberte oblast použití teorie max-flow min-cut teorému a toto téma zpracujte. Dále viz

zadání Rešerše.

Literatura:

- T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 2002.
- Tom Leighton, Satish Rao: *Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms*, Journal of the ACM (JACM), Volume 46 Issue 6, Nov. 1999, pages 787 – 832. [<http://dl.acm.org/citation.cfm?id=331524.331526>]
- S. M. Sadegh Tabatabaei Yazdi, Serap A. Savari: *A max-flow/min-cut algorithm for a class of wireless networks*, SODA '10 Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1209-1226. [<http://dl.acm.org/citation.cfm?id=1873601.1873698>]
- Serge A. Plotkin, Éva Tardos: *Improved bounds on the max-flow min-cut ratio for multicommodity flows*, STOC '93 Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, pages 691-697. [<http://dl.acm.org/citation.cfm?id=167088.167263>]

Efektivní algoritmy pro hledání maximálního párování

Viz zadání Rešerše.

Literatura:

- Zvi Galil: *Efficient Algorithms for Finding Maximum Matching in Graphs*. *Computing Surveys*, Vol. 18, No. 1, p. 23-38, 1986.

Planární grafy a jejich vlastnosti

Viz zadání Rešerše.

Literatura:

- Alan Gibbons: *Algorithmic Graph Theory*, Chapter 3: Planer graphs, p. 67-95. Cambridge University Press, 1985.

Teorie k přiřazovacím problémům

Viz zadání Rešerše.

Literatura:

- R. Burkard et al.: *Assignment Problems*, Chapter 2: Theoretical foundations, p. 13-34. SIAM, 2009..

Algoritmy pro bipartitní párování

Viz zadání Rešerše.

Literatura:

- R. Burkard et al.: *Assignment Problems*, Chapter 3: Bipartite matching algorithms, p. 35-71. SIAM, 2009.

Permutační směrování v sítích založených na mřížce

Viz zadání Rešerše.

Literatura:

- Permutation routing in mesh-based networks (Section #9 in CS838: Topics in parallel computing) [poznámky k přednášce], Dostupné na

<https://www.fit.vutbr.cz/study/courses/GAL/private/materials/permutation-routing.pdf>.

(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, ...).

Algoritmus pro určení hranové konektivity grafu

Viz zadání Řešení paralelizace... .

Literatura:

- H. Nagamochi, T. Ibaraki, Computing edge-connectivity in multigraphs and capacitated graphs. *SIAM J. Discrete Math.* **5**, p. 54-66, 1992.
- J. A. Bondy, U. S. R. Murty, Graph Theory, Springer, 2008. Chapter 9.

Egerváryho algoritmus (párování v bip. grafech)

Viz zadání Řešení paralelizace... .

Literatura:

- J. A. Bondy, U. S. R. Murty, Graph Theory, Springer, 2008. Chapter 16.

Dijkstrův algoritmus (hledání nejkratších cest)

Viz zadání Řešení paralelizace... .

Literatura:

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

(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 srovnajte vhodnost těchto algoritmů pro různé typy grafů (řídce/husté grafy, ...). Dosažené výsledky zdůvodněte. Implementace a porovnání více než dvou algoritmů může být hodnoceno bonusovými body. Grafické uživatelské rozhraní je pouze volitelné (standardně se počítá s tím, že výstupem projektu tohoto typu bude konzolová aplikace).

Porovnání dvou algoritmů: Hledání párování maximální kardinality

Viz zadání Porovnání dvou algoritmů.

Literatura:

- Rainer Burkard et al.: *Assignment Problems*, Chapter 3. SIAM, 2009

Porovnání dvou algoritmů: Vyhledání silně souvislých komponent (Gabowův algoritmus a Tarjanův algoritmus)

Viz zadání Porovnání dvou algoritmů.

Literatura:

- http://en.wikipedia.org/wiki/Path-based_strong_component_algorithm#References
- R. E. Tarjan, Depth-first search and linear graph algorithms. *SIAM Journal on Computing* 1 (2), p. 146-160, 1972.

Porovnání dvou algoritmů: Vyhledání silně souvislých komponent (Kosarajův algoritmus a Gabowův algoritmus)

Viz zadání Porovnání dvou algoritmů.

Literatura:

- T.H. Cormen, C.E. Leiserson, R.L. Rivest, *Introduction to Algorithms*, McGraw-Hill, 2002.
- http://en.wikipedia.org/wiki/Path-based_strong_component_algorithm#References

Porovnání dvou algoritmů: Testování planarity grafu

Viz zadání Porovnání dvou algoritmů.

Literatura:

- A. Gibbons: *Algorithmic Graph Theory*, Section 3.4. Cambridge University Press, 1985.
- J. Hopcroft, R. E. Tarjan, Efficient planarity testing. *J. Assoc. Comput. Mach.* 21(4), p. 549-568, 1974.

- http://en.wikipedia.org/wiki/Planarity_testing#References

Porovnání dvou algoritmů: Alokace registrů pomocí barvení

Viz zadání Porovnání dvou algoritmů.

Literatura:

- Briggs, P., Cooper, K.D., Kennedy, K., Torczon, L.: *Coloring heuristics for register allocation*, PLDI '89 Proceedings of the ACM SIGPLAN 1989 Conference on Programming language design and implementation, pages 275 – 284. [<http://portal.acm.org/citation.cfm?id=74843>]
- Briggs, P., Cooper, K.D., Torczon, L.: *Improvements to graph coloring register allocation*, ACM Transactions on Programming Languages and Systems (TOPLAS), Volume 16 Issue 3, May 1994, pages 428 – 455. [<http://portal.acm.org/citation.cfm?id=177492.177575>]
- *Tailoring Graph-coloring Register Allocation For Runtime Compilation* [online], [cit. 27. 9. 2014], <http://llvm.org/pubs/2006-04-04-CGO-GraphColoring.html>

(E) Speciální zadání

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

Generování grafů různých vlastností dle zadaných parametrů

Implementační zadání kombinované s řešením existujících nástrojů pro generování grafů vhodných pro různé testování. Implementace bude potom vytvářet generátor, který mezi existujícími nástroji nebyl dostupný (výběr konzultujte se cvičícím). Aplikace bude konzolová a bude umožňovat různou parametrizaci a výstup ve vybraném standardním formátu (např. GraphML či jiné běžné formáty). Jako bonus se pokuste o paralelizaci generování.

Literatura:

- Doporučujeme vyhledávat nejen na webu, ale i databázích odborných článků.

Vícekriteriální optimalizace ohodnoceného neorientovaného grafu

Pro zadanou množinu vrcholů V , vzdálenostní funkci $dist: V \times V \rightarrow \mathbb{N}$, funkci historie $h: V \times V \rightarrow \{0, b^0, b^1, b^2\}$ (b je parametr) a funkci požadovaného stupně $deg_{req}: V \rightarrow \{1, 2, \dots, m\}$ (m je další parametr) vytvořte/vygenerujte vždy alespoň jeden souvislý neorientovaný graf $G = (V, E)$ takový, že (podmínky jsou řazeny od nejdůležitější po méně důležitou):

1. pro každé $v \in V$ platí $deg_{req}(v) = deg(v)$;
2. pro všechny $(u, v) \in E$ platí, že $h(u, v)$ je minimální;
3. pro všechny $(u, v) \in E$ platí, že $dist(u, v)$ je minimální.

Navrhněte 2-3 optimalizační techniky pro generování grafu G a po konzultaci s cvičícím je implementujte. Provedte experimenty a studujte plnění podmínek a implementované algoritmy porovnejte a zhodnoťte. Jako bonus se pokuste o paralelizaci generování.