VŠB TECHNICKÁ UNIVERZITA OSTRAVA
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Komponenta výukového serveru TI NP-úplné problémy 2

Component of Learning Server for
Theoretical Computer Science
NP-complete problems 2

Bakalářská práce

Phat Tran Dai

Vedoucí práce: Ing. Martin Kot, Ph.D.

Ostrava, 2025

VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Č.j.VSB/25/140522
vsboee25040F4E

Zadání bakalářské práce

Student: **Phat Dai Tran**

Studijní program: B0613A140014 Informatika

Téma: Komponenta výukového serveru TI - NP-úplné problémy 2

Component of Learning Server for Theoretical Computer Science - NP-complete problems 2

Jazyk vypracování: čeština

Zásady pro vypracování:

V rámci diplomových a bakalářských prací vzniká výukový server pro předměty teoretické informatiky. Jedná se o sadu dynamických webových stránek umožňujících studentům pochopení různých typů úloh a problémů tím, že si mohou zadat na stránce libovolné zadání a zobrazí se jim řešení včetně postupu. Cílem této práce je vytvořit komponentu (tedy sadu webových stránek) pro výuku vybraných NP-úplných problémů a převodů mezi nimi.

1. Nastudujte si problematiku tříd složitosti problémů a s tím souvisejícím převodem mezi problémy.
2. Vytvořte dynamické webové stránky umožňující uživateli následující:
a) Nechat si zobrazit postupně po krocích postup algoritmu s polynomiální časovou složitostí převádějícího zadanou instanci jednoho problému na instanci jiného problému (budou implementovány alespoň 3 různé převody mezi problémy).
b) Zadat libovolnou instanci každého z problémů vyskytujících se v těchto převodech.
c) Zobrazit odpověď na otázku daného problému pro zadanou instanci, v případě kladné odpovědi i se zdůvodněním.
3. Není cílem mít co nejefektivněji implementován samotný převod, ale mít jej implementován tak, aby uživateli byla myšlenka tohoto převodu co nejsrozumitelněji ukázána.
4. Vytvořte i ukázkové vstupní instance pro implementované problémy tak, aby uživatel mohl vše vyzkoušet i bez zadávání vlastních vstupů (alespoň 5 instancí pro každý problém).

Studenti řešící toto zadání s rozdílným číslem v názvu mohou (ale nemusí) spolupracovat tak, že výsledek může mít společné uživatelské rozhraní apod. Ale každý bude implementovat jiné 3 převody mezi problémy.

Seznam doporučené odborné literatury:

[1] Sipser, M.: Introduction to the Theory of Computation, PWS Publishing Company, 1997.
[2] Papadimitriou, C.: Computational Complexity, Addison Wesley, 1993.
[3] Sawa, Z.: Prezentace přednášek předmětu Teoretická informatika, dostupné online https://www.cs.vsb.cz/sawa/ti/index.html

17. listopadu 2172/15
708 00 Ostrava-Poruba
Česká republika

spojovatelka: +420 597 321 111
epodatelna: epodatelna@vsb.cz
ID datové schránky: d3kj88v

IČ: 61989100
DIČ: CZ61989100

email: studijni.fei@vsb.cz
www.fei.vsb.cz

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Martin Kot, Ph.D.**

Datum zadání: 01.09.2025

Datum odevzdání: 30.04.2026

Garant studijního programu: Ing. Tomáš Fabián, Ph.D.

V IS EDISON zadáno: 14.10.2025 13:16:02

17. listopadu 2172/15
708 00 Ostrava-Poruba
Česká republika

spojovatelka: +420 597 321 111
epodatelna: epodatelna@vsb.cz
ID datové schránky: d3kj88v

IČ: 61989100
DIČ: CZ61989100

email: studijni.fei@vsb.cz
www.fei.vsb.cz

Abstrakt

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At.

Klíčová slova: Lorem, ipsum, dolor, sit, amet„ consectetur, adipiscing, elit„ sed, do, eiusmod, tempor, incididunt, ut, labore, et, dolore, magnam, aliquam, quaerat.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.

Keywords: Lorem, ipsum, dolor, sit, amet„ consectetur, adipiscing, elit„ sed, do, eiusmod, tempor, incididunt, ut, labore, et, dolore, magnam, aliquam, quaerat.

Poděkování

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri.

Seznam použitých zkratek a symbolů

GCC	– GNU C Compiler
GNU	– GNU Is Not Linux
NPTIME	– Non-Deterministic Polynomial Time

Obsah

Úvod

Uvést čtenáře do kontextu.
– Vysvětlit stručně teorii složitosti algoritmů a v průběhu se zaměřit na termín NP--Complete.

Jaká je motivace téhle práce? Co je problémem, který se snaží vyřešit?
– Studenti a vyučující TI, chtějí vizualizovat převody mezi problémy.
– Z vizualizací lze mnohem pochopit redukce, než ze čtení formálních textů.
– Umožnit interaktivní zobrazení redukcí a jejich postup.
– Zobecnění pro, každou instanci problému. Konec statických animací redukcí pro specifické vstupní instance.

Jaké cíle má tahle práce? Co chce přínést, unsnadnit, v čem má pomoct?
– Přijít s řešením pro satisfakci vylistovaných motivací.
– Má za cíl, poskytnout interaktivní GUI pro zadavaní instancí.
– Vizualizace vstupních a výstupních instancí, ať už to jsou logické výrazy, orientované/ neorientované grafy s nebo bez ohodnocení hran, tabulky pro čísla jako vstupní instance pro problém SSP.
– Zobrazit certifikáty k daným problémum, pokud existují.
– Zobrazit i certifikát pro vstupní instanci, pokud má certifikát výstupní instance.
– Zobrazit postup pro jednotlivé redukce.

Co bylo vytvořeno.
– Webová aplikace, která funguje podobně jako online kalkulačky (např. integrální kalkulačky, zobrazující postup, atd…).

Kapitola 1
Základní teorie

Než se vrhnem přímo do věci, je dobré si probrat základní teorii, bez které se ve čtení této práce neobejdeme.

Příponemene si co je to algoritmus, rozhodovací problém, složitost algoritmu, třídy složitosti jako jsou P, NP, NP-těžké a NP-úplné problémy.

1.1 Problém a algoritmus

Algoritmy můžeme chápat jako instrukce pro vyřešení nějakého problému. To, co algoritmus je, lze definovat mnoha způsoby a neexistuje přesná definice. Problém je to, co lze řešit, což ale nutně neznamená, že lze vyřešit.

Algoritmus A řešící problém P znamená to, že pro každou instanci problému P dostaneme aplikovaním algoritmu A vždy správnou odpověď.

Např. pro třídicí problémy by tato odpověď byla setřízená sekvence. Pro algoritmus `f(a) = a * a` bychom získali druhou mocninu čísla `a`. Pro algoritmus `isCapital(a: char) -> Bool` vratí True pro chary A-Z a False pro a-z.

1.2 Rozhodovací problém

Pro jednoduchost se obvykle omezujeme na tvz. rozhodovací problémy. Rozhodovací problémy je trojice (I, O, Q), kde I je množina správných vstupu, O je množina možných výstupu a Q je otázka. Množina O obsahuje dvě položky, *Ano* a *Ne*, které představují odpověď na otázku Q.

Každý problém lze zformulovat do rozhodovacího problému, nehledě na to zda se jedná o optimalizační, třídicí či jakýkoliv jiný problém.

Mějme např. problém vyhledavání nejkratší cesty v ohodnoceném neorientovaném grafu, který představuje optimalizační problém (konkrétně minimalizační problém). Problémem je najít nejkratší cestu. Tento problém lze transformovat do rozhodovacího problému tím, že zavedeme konstantu $k \in \mathbb{R}$ jako další položku vstupu a zeptáme se otázku: Existuje v grafu cesta z vrcholu a do b, taková, že její cena je menší nebo rovna k?

1.3 Výpočetní model

Instrukcemi algoritmů můžeme chápat jako kroky v Turingových strojích, které představují teoretickou abstrakci výpočetního stroje. Říkáme, že Turingův stroj je výpočetní model a není zdaleka jediným.

Výpočetních modelů existuje spousta a jsou jimi např. páskové stroje, zásobníkové automaty (s jedním, dvěma či více zásobníky), formální jazyky, konečné stavové automaty, stoje s jedním číslem (nevím jak se jmenuje, sawa ma v prez) a mnoho dalších.

Avšak hlavním rozhodujicím rozdílem v jejích výpočetní schnoponosti (síle) je, zda jsou tvz. Turing-úplné. Turing-úplné modely jsou takové výpočetní modely, kterými lze modelovat všechny algoritmy. Modely, které tuto vlastnost nemají, jsou např. konečné automaty, formální jazyky a zasobníkové automaty s jedním zásboníkem. Pokud by zasobníkový automat měl dva zásobníky, byl by Turing-úplný.

1.4 Algorimická řešitelnost problému

Ne každý problém je řešitelný algoritmem. Těm problémům, které nelze vyřešit algoritmem, říkáme algoritmicky neřešitelné. Nejznámnější z nich je problém zastavení, ang. Halting Problem, a tázá se: je-li dán algoritmus A, vykoná se algoritmus A v konečném počtu kroků? Protože otázka tohoto problému zní velmi jednodušše, člověka by mohlo napadnout, že nalézt odpověď na ní, bude velmi jednoduché. Avšak po lehké úvaze dojdeme k tomu, že se zde nachází paradox.

Předpokládejme, že pro tento problém existuje algoritmus `fn halts(f) -> bool`, který ho řeší.

```
fn halts(f) -> bool {
  if f halts after finite amount of steps {
    return True
  }
  else {
    return False
  }
}

fn h() {
  if halts(h) {
    while True {
      continue
    }
  }
}
```

Pokud algoritmus H (funkce) je algoritmus, který se zastaví, výraz vrátí `halts(H)` vrátí True a funkce H se začne vykonávat nekonečnou smyčku, tedy funkce H se nezastaví, pokud se zastaví. Je zde kontradikce, proto funkce `halts` nemůže existovat.

1.5 Asymptotická notace

Teorie složistosti studuje časovou a prostorovou (paměťovou) složitost algoritmů. Časovou složitostí algoritmu A se rozumí to, jak rychle poroste počet kroků (instrukcí), které se vykonají, s velikostí vstupu algoritmu A.

Měřítek složitosti máme několik, nejčastější používáme $O(f)$, $\Omega(f)$ a $\Theta(f)$, kde f je nějaká funkce $f : \mathbb{N} \to \mathbb{R}$, a používáme je ve smysle v nejhorším případě, v nejlepším případě a v průměrném případě v tomto pořadí. Obecně jsou známy pod názvem asymptotická notace. Měřítko $O(f)$ značí horní hranici, $\Theta(f)$ dolní hranici a $\Omega(f)$ je mezi dolní a horní hranicí počtů kroků s rostoucí velikostí vstupní instance.

Formálněji, mějme funkce f, g a h, kde jsou všechny předpisu $\cdot : \mathbb{N} \to \mathbb{R}$. Pokud funcke přiřadíme do notací tímto způsobem:

$$O(f)$$
$$\Theta(g)$$
$$\Omega(h)$$

potom musí platit pro $x \in \mathbb{R}$, že:

$$f(x) \geq g(x)$$
$$h(x) \geq g(x)$$
$$h(x) \leq f(x)$$

To, že o algoritmu řekneme že má složitost $O(f)$, znamená, že v nejhorším případě počet kroků nepřesáhne $k \cdot f(x)$ v závislosti na velikosti vstupu x, kde $k \in \mathbb{R}$ (nebude horší než). Je to tedy omezení zhora.

$$\text{počet kroků} \leq k \cdot f(x)$$

Složitost $\Omega(g)$ znamená, že v nejlepším případě nebude počet kroků menší než $k \cdot g(x)$ (nebude lepší než) Jedná se o omezení zdola.

$$k \cdot f(x) \leq \text{počet kroků}$$

Složitost $\Theta(h)$ znamená pro $k_1, k_2 \in \mathbb{R}$:

$$k_1 \cdot h(x) \leq \text{počet kroků} \leq k_2 \cdot h(x)$$

Jedná se o omezení jak zdola, tak zhora a lze většinou ho chápat jako měřítko pro průměrný případ.

S pojmem časová složitost se pojí pojem prostorová složitost. Ta studuje složitost v rámci počtu paměťových buněk potřebné k vykonání algoritmu. Také se zde používá asymptotická notace, tady avšak nehledíme na počet kroků ale počet paměťových buněk.

Jeden ze známých třídicíh algoritmů je `bubblesort`.

```
1  def bubblesort(arr: list[int]) {
2    for i in range(0, len(arr)):
3      for j in range(i + 1, len(arr)):
4        if arr[i] > arr[j]:
5          arr[i], arr[j] = arr[j], arr[i]
```

Tento algoritmus má složitost $O(n^2)$, $\Theta(n^2)$ a $\Omega(n^2)$. V nejhorším, nejlepším a v průměrném případě má složitost n^2.

Dalším daleko efektivnějším třídicím algoritmem je `quicksort` se složitostí $O(n^2)$, $\Omega(n\log(n))$ a $\Theta(n\log(n))$.

```
1  quicksort [] = []
2  quicksort (p:xs) = (quicksort lesser) ++ [p] ++ (quicksort greater)
3    where
4      lesser = filter (< p) xs
5      greater = filter (>= p) xs
```

1.6 Třídy složitosti

Problémy lze na základě jejich časové a prostorové složitosti klasifikovat do tzv. tříd složitostí. Zde budeme uvažovat pouze o časové složitosti. Některé z těchto tříd jsou L, P, EXP, NP, NL.

– L - množina problémů s deterministickým algoritmem se složitostí $\log(n)$
– P - množina problémů s deterministickým algoritmem se složitostí n^k, kde $k \in \mathbb{R}$
– EXP - množina problémů s deterministickým algoritmem se složitostí k^n, kde $k \in \mathbb{R}$
– NP - množina problémů s nedeterministickým algoritmem se složitostí n^k, kde $k \in \mathbb{R}$
– NL - množina problémů s nedeterministickým algoritmem se složitostí $\log(n)$

U definic nedeterministických tříd N- existuje alternativní definice.

Třídu NP tak lze definovat taktéž jako množinu rozhodovacích problému, u kterých existuje-li kladná odpověď, tak jejich certifikát (taktéž svědek nebo důkaz) lze ověřit v polynomiálním čase.

Z těchto definic jsou zřejmé, že:

$$L \subset P \subset \text{EXP}$$
$$L \subset \text{NL}$$
$$P \subseteq \text{NP nebo } P \subset \text{NP}$$

Všimněte si, že pro třídu P a NP není znám přesný vztah. Zda třída P se rovná či nerovná třídě NP je otevřená otázka. Důkaz, který by vyvrátil nebo zdůvodnil ono nebo druhé, ještě nebyl nikdy zformulován.

1.7 Redukce

Redukce problému A na problém B znamená převedení instance problému A na instanci problému B tak, že z odpověď na problém B můžeme vykonstruovat odpověď pro instance problému A.

Jsou-li problémy A a B rozhodovací problémy, odpověď instance problému B se musí rovnat odpovědi problému A. Musí se zachovat kladná i záporná odpověď.

Můžeme jej chápat jako problém, který je řešen nějakým algoritmem. Skoro vždy nám ale jde o to, aby vzniklá instance problému B nevzrostla exponencionálně od velikosti instace problému A. Budeme tedy brát v potaz pouze algoritmy se složitostí $O(n^k)$.

1.8 Třída NP

Ve třídě NP jsou takové problémy, pro které umíme řešit s časovou složitostí $O(k^n)$ deterministickým algoritmem. Pokud by se vynalezl stroj, který by uměl pracovat nedeterministickými instrukcemi, dokázali bychom tyto problémy vyřešit v polynomiálním čase. Zatím ale nikdo takový stroj nevynalezl.

S třídou NP jsou spojené třídy NP-těžkých a NP-úplných problémů. Problém T nazveme NP-těžkým, jestliže lze na něj redukovat každý problém N ve třídě NP. Problém U nazveme NP-úplným, pokud je NP-těžký a zároveň patří do třídy NP.

NP-úplné problémy jsou z definice takové, které lze redukovat mezi sebou. Z důsledku, nalezneme-li efektivní algoritmus pro jeden NP-úplný problém, nalezli jsme efektivní algoritmus pro každý NP-úplný problém.

Závěr

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere.

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et.

Literatura

[1] PREKAS, George, Marios KOGIAS a Edouard BUGNION. ZygOS: Achieving Low Tail Latency for Microsecond-Scale Networked Tasks. In: *Proceedings of the 26th Symposium on Operating Systems Principles* [online]. B.m.: Association for Computing Machinery, 2017, s. 325–341. Dostupné z: doi:10.1145/3132747.3132780

[2] MEHTA, Jiten a Eric KINNEAR. *Boost Performance and Security with Modern Networking* [online]. 26. červen 2020 [cit. 2020-09-17]. Dostupné z: https://developer.apple.com/videos/play/wwdc2020/10111/

[3] OMAROVA, Saule a Graham STEELE. There's a Lot We Still Don't Know About Libra. *The New York Times* [online]. 2019. Dostupné z: https://www.nytimes.com/2019/11/04/opinion/facebook-libra-cryptocurrency.html

[4] DONNE, John. *The "Anniversaries" and the "Epicedes and Obsequies"*. B.m.: Indiana University Press, 1995. The Variorum Edition of the Poetry of John Donne.

[5] BROWN, George C., ed. A Swedish Traveller in Early Wisconcin: The Observations of Fredrika Bremer. *Wisconcin Magazine of History*. 2. vyd. 1978, **61–62**.

[6] In: J. K. ROWLING*Harry Potter and the Order of the Phoenix*. cca. 2003, s. 135–139.

[7] MÄDJE, Laurenz. *Tokenization of + and - with scientific notation* [online]. 18. červenec 2020. Dostupné z: https://github.com/typst/typstc/issues/3

[8] *Terminator 2: Judgment Day*. Carolco Pictures; Pacific Western Productions; Lightstorm Entertainment; Le Studio Canal+ S.A. 1. červenec 1991.

[9] Conspiracy Theories and Interior Design. In:. Universal Television; Sony Pictures Television; Krasnoff Foster Productions; Harmonious Claptrap; Russo Brothers Film. 18. listopad 2010.

[10] *The wire*. Blown Deadline Productions. 2002.

[11] DOAN, T. D., D. B. TRAN THOAI a Hartmut HAUG. Kinetics and luminescence of the excitations of a nonequilibrium polariton condensate. *Physical Review B* [online]. 2020, **102**(16), 165126–165139. Dostupné z: doi:10.1103/PhysRevB.102.165126

[12] JERRENTRUP, Andreas, Tobias MUELLER, Ulrich GLOWALLA, Meike HERDER, Nadine HENRICHS, Andreas NEUBAUER a Juergen R. SCHAEFER. Teaching medicine with the help of "Dr. House". *PLoS ONE* [online]. 2018, **13**(3). Dostupné z: doi:10.1371/journal.pone.0193972

[13] *Informational plaque about Jacoby's 1967 photos*. B.m.: Stiftung Reinbeckhallen. 2020

[14] *L'oiseau rare, de l'hirondelle au kakapo* [online]. 18. prosinec 2020 [cit. 2020-11-04]. Dostupné z: https://www.museedesconfluences.fr/fr/evenements/l%E2%80%99oiseau-rare-de-l%E2%80%99hirondelle-au-kakapo

[15] DUVAL, Fred. *Renaissance, Les Déracinés*. 1. vyd. Přel. EMEM a Fred BLANCHARD. B.m.: Dargaud, 2018.

[16] MOORE, Edward F. *Gedanken-experiments on sequential machines*. B.m.: NBS. duben 1956. Annals of Mathematics Studies

[17] SILVER, Nate. Trump's claim to have won Georgia is highly dubious. No network has called it. He's only ahead by 2.5 points there, and the outstanding votes are mostly mail votes in very blue counties, likely very Democratic. Biden may even be a slight favorite there. In: [online]. 4. listopad 2020. Dostupné z: https://twitter.com/NateSilver538/status/1323889051037028353

[18] PEDBOST, Marven F., Trillean POMALGU, Chris LINTOTT, Nora EISNER a Belinda NICHOLSON. *Defining the Really Habitable Zone* [online]. 2020. Dostupné z: https://arxiv.org/abs/2003.13722

[19] ISHKUR. *Ishkur's Guide to Electronic Music* [online]. [cit. 2020-11-12]. Dostupné z: http://www.techno.org/electronic-music-guide/

[20] MATTERMOST. Mattermost Privacy Policy. *Policies* [online]. [cit. 2020-11-29]. Dostupné z: https://mattermost.com/privacy-policy/

[21] WORTH, Jon. *Jon Worth Euroblog* [online]. Dostupné z: https://jonworth.eu/

[22] PROKOPOV, Nikita. It is fast or it is wrong. *tonsky.me* [online]. 29. prosinec 2018. Dostupné z: https://tonsky.me/blog/slow-wrong/

[23] UNITED NATIONS DEVELOPMENT PROGRAMME. *Human Development Report 2019* [online]. 2019. Dostupné z: http://hdr.undp.org/sites/default/files/hdr2019.pdf

[24] BARROWS, Miellyn Fitzwater. Audio Descriptions. In: [online]. 7. únor 2017. Dostupné z: https://www.20k.org/episodes/audio

[25] STEYERL, Hito. *Drill*. 20. červen 2019

[26] *authoritative* [online]. nedatováno [cit. 2020-11-29]. Dostupné z: https://dictionary.cambridge.org/dictionary/english/authoritative

[27] *Logician* [online]. nedatováno [cit. 2019-12-02]. Dostupné z: http://image-net.org/api/text/wordnet.structure.hyponym?wnid=n10269785

[28] INTERNET ENGINEERING TASK FORCE. *Secret Key Transaction Authentication for DNS* [online]. 2000. Dostupné z: https://tools.ietf.org/html/rfc2845

[29] *Roe v. Wade*. 1973

[30] *Freedom of Information Act*. 1967

[31] JOHN. Celebrating over five million users, a quarter million daily actives, and over five years of dedicated user support. *Overleaf Blog* [online]. 8. listopad 2019. Dostupné z: https://de.overleaf.com/blog/celebrating-over-five-million-users-and-a-quarter-million-daily-actives

[32] PEPE, Alberto. *How many scholarly articles are written in LaTeX?* [online]. 21. únor 2017. Dostupné z: doi:10.22541/au.148771883.35456290

[33] MUNROE, Randall. *Types of Editors* [online]. 12. březen 2014. Dostupné z: https://xkcd.com/1341/

[34] GÜNTHER-HAUG, Barbara. *Den Boden unter den Füßen verlieren.* B.m.: MVG, 2020.

Příloha A

Demo appendix

Look at this quicksort algorithm written in Haskell.

```
quicksort [] = []
quicksort (p:xs) = (quicksort lesser) ++ [p] ++ (quicksort greater)
  where
    lesser = filter (< p) xs
    greater = filter (>= p) xs
```

Very cool.