VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

Slovník

Semestrální práce

Autor práce: Aleksandr Shabelnikov

Jihlava 2022

Obsah

1 Úvod

Algoritmy a datové struktury jsou základními pojmy informatiky, které poskytují základ pro navrhování a implementaci efektivních řešení výpočetních problémů. Algoritmy jsou souborem instrukcí, které se používají k řešení konkrétního problému, zatímco datové struktury slouží k uspořádání a uložení dat způsobem, který umožňuje efektivní přístup k nim a jejich úpravu.

Dobrá znalost algoritmů a datových struktur je nezbytná pro každého, kdo se snaží o kariéru v oblasti informatiky nebo v příbuzných oborech. Je to proto, že mnoho problémů, s nimiž se informatici při své práci setkávají, zahrnuje vývoj a analýzu algoritmů a návrh a implementaci efektivních datových struktur. Kromě toho je porozumění těmto pojmům nezbytné pro pochopení pokročilejších témat v informatice, jako je umělá inteligence, strojové učení a počítačové sítě.

Stručně řečeno, studium algoritmů a datových struktur je nezbytnou součástí vzdělání v oblasti informatiky, protože poskytuje základní znalosti a dovednosti, které jsou nezbytné pro úspěch v oboru.

2 Binární vyhledávací stromy

Binární vyhledávací strom je typ datové struktury, která se používá k hierarchickému ukládání dat tak, že každý uzel stromu má nejvýše dva potomky. Data v každém uzlu jsou uspořádána tak, aby je bylo možné rychle vyhledávat a přistupovat k nim.

Jednou z hlavních výhod použití binárního vyhledávacího stromu je, že umožňuje rychlé vyhledávání a třídění dat. Strom je totiž uspořádán tak, že k datům lze rychle přistupovat tak, že se začne v kořenovém uzlu a poté se prochází stromem na základě hodnoty hledaného údaje. Pokud je například hledaná hodnota menší než hodnota aktuálního uzlu, lze se přesunout na levého potomka aktuálního uzlu; pokud je hledaná hodnota větší než hodnota aktuálního uzlu, lze se přesunout na pravého potomka aktuálního uzlu. Tento proces pokračuje, dokud není nalezen požadovaný údaj nebo dokud není zjištěno, že údaj ve stromu neexistuje.

Binární vyhledávací stromy mají řadu reálných aplikací. Například se běžně používají v databázích k uspořádání a uložení dat způsobem, který umožňuje jejich efektivní vyhledávání a získávání. Používají se také v počítačových programech k implementaci slovníků a jiných datových struktur, které vyžadují rychlé vyhledávání a třídění. Kromě toho se binární vyhledávací stromy často používají v oblasti informatiky k vývoji a analýze algoritmů pro řešení výpočetních problémů.

Pro implementaci česko-anglického a anglicko-českého slovníku jsem zvolil právě BST.

3 Zadání

Napište program realizující slovník, ve kterém českému slovu (bez diakritiky) odpovídá jedno nebo několik anglických slov. Počáteční obsah slovníku bude dán textovým souborem. Program pak umožňuje slovník doplňovat, opravovat a v něm hledat.

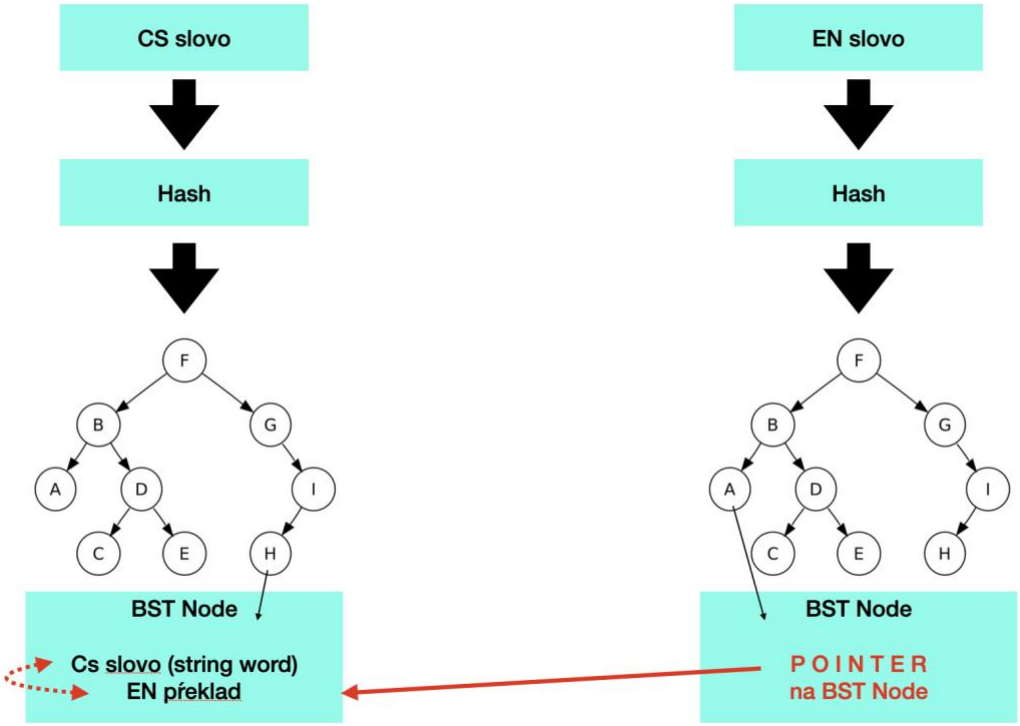
Operace:

- najít anglické překlady k zadanému slovu
- zrušit záznam pro zadané slovo
- vložit nový překlad nebo i nové slovo k překladu
- vypsat slovník

Při práci na projektu jsem se rozhodl rozšířit minimální funkčnost o následující funkce:

- Oboustranné překlady
- Uložení anglicko-českého slovníku do souboru
- Uložení česko-anglického slovníku do souboru

4 Realizace



Při realizaci projektu bylo rozhodnuto použít 2 binární stromy: jeden pro anglický a druhý pro český slovník. Zvláštností je, že strom pro anglický slovník neukládá slova, ale pouze ukazatel na uzel, který obsahuje obě slova (anglické i české).

4.1 Obecný princip aplikace

1. Načtení souboru CSV cs-en.csv, který obsahuje 2000 řádků českých a anglických slov
2. Použití hashovacího algoritmu ze standardní knihovny STL na slovo pro získání klíče. Výsledkem třídy std::hash je proměnná typu size_t (její velikost v paměti závisí na operačním systému. Například v systému MacOS Ventura na procesorech ARM je to 8 bajtů, což odpovídá velikosti a typu proměnné unsigned long). To je klíč k nalezení dat.
3. Operace se dvěma stromy (vyhledávání, mazání, přidávání)
4. Výsledek

4.2 Popis souboru projektu

- BST.cpp, BST.h — Soubory třídy BST. Hlavní funkce programu.
- BST_En.cpp, BST_En.h — Pomocné soubory třídy BST_En pro práci s anglickými slovy.
- Node.h, Node_En.h — Třídy pro práci s uzly stromu.
- Main.cpp — Uživatelské rozhraní.

Zdrojový kód je součástí archivu spolu s dokumentací.

4.3 Odůvodnění používání ukazatelů

Ukazatele v jazyce C++ umožňují přímou manipulaci s adresami paměti, což může být efektivnější než ukládání stejného objektu jako řetězce Unicode. Například při použití ukazatelů můžete přímo měnit hodnotu uloženou na paměťové adrese, zatímco v případě řetězce Unicode byste museli vytvořit nový objekt řetězce a zkopírovat hodnoty z toho starého. To může ušetřit čas a paměť, zejména při práci s velkými objekty nebo datovými sadami. Použití ukazatelů navíc umožňuje vytvářet flexibilnější a výkonnější programy, protože máte přímý přístup k adresám paměti.

Porozumění ukazatelům je v informatice důležité z několika důvodů. Za prvé, ukazatele poskytují způsob, jak přímo manipulovat s adresami paměti, což může být mocný nástroj pro vytváření efektivních a účinných programů. Za druhé, mnoho programovacích jazyků, včetně jazyka C++, používá ukazatele jako základní koncept, takže porozumění ukazatelům je pro práci s těmito jazyky nezbytné. Zatřetí, pochopení ukazatelů vám pomůže porozumět tomu, jak počítače a softwarové systémy fungují na základní úrovni, což může být užitečné pro různé úlohy, například ladění a optimalizaci kódu. A konečně, pochopení pointerů vám může také pomoci hlouběji porozumět dalším konceptům informatiky, jako je správa paměti a datové struktury.

5 Závěr

Ukázalo se, že aplikace slovníku není optimální, není ideální a má řadu omezení. Neumí si poradit s kolizí a není možné zadat více než jeden překlad pro jedno slovo. Jako školicí projekt zvládá své funkce a funguje podle zadání.

Je těžké posoudit, jak optimální je BST pro řešení zpracování textu, stromy fungují dobře pro vyhledávání v indexech, ale v reálném životě stálo za to vyzkoušet implementaci projektu na různých typech stromů, jako jsou AVL nebo černo-červené, vážené a samooptimalizující.

Pro přidání více překladů by bylo vhodné použít složitější slovníkový uzel, např. ukazatel na spojový seznam, který by pro daný hash index uchovával všechny možné překlady.

Projekt je vytvořen v prostředí CLion IDE a byl zkompilován pro operační systémy MacOS a Linux. Bohužel nemám počítač s Microsoft Windows, takže při kompilaci v tomto operačním systému mohou (ale neměli by) nastat problémy s řetězci Unicode.

6 Příloha. Snímky obrazovky

```
mrmidi@collier cmake-build-debug % uname -a
Darwin collier.ad.vspj.cz 22.2.0 Darwin Kernel Version 22.2.0: Fri Nov 11 02:04:
44 PST 2022; root:xnu-8792.61.2~4/RELEASE_ARM64_T8103 arm64
mrmidi@collier cmake-build-debug % ./SEM
---------------
| Dictionary   |
---------------
Semestral project for D S A
2022, Aleksandr Shabelnikov, VSPJ

Loading data from file
Opening file /Users/mrmidi/CLionProjects/SEM/cs-en.csv
Loaded 2000 words
Time: 0.004718
1. Add new word
2. Search czech word
3. Search english word
4. Delete word
5. Print dictionary
6. Make new file with cs-en dictionary
7. Make new file with en-cs dictionary
8. Exit
Enter your choice:
```

```
mrmidi@mrmidi:~/dsa/SEM/build$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
mrmidi@mrmidi:~/dsa/SEM/build$ ./SEM
---------------
| Dictionary   |
---------------
Semestral project for D S A
2022, Aleksandr Shabelnikov, VSPJ

Loading data from file
Opening file /home/mrmidi/dsa/SEM/cs-en.csv
Loaded 2000 words
Time: 0.006162
1. Add new word
2. Search czech word
3. Search english word
4. Delete word
5. Print dictionary
6. Make new file with cs-en dictionary
7. Make new file with en-cs dictionary
8. Exit
Enter your choice:
```

Spuštění programu v systému linux a macOS

```
Enter your choice: 2
Enter word: zkouška
Word not found
Time: 7.9e-05
```

```
Enter your choice: 1
Enter word: zkouška
Enter definition: exam
```

```
Enter your choice: 2
Enter word: zkouška
Word: zkouška
Definition: exam
Node address: 0x556c14c559b0
Time: 0.000166
```

```
Enter your choice: 3
Enter word: exam
Node address: 0x556c14c559b0
Word: exam
Definition: zkouška
Time: 0.00016
```

Přiklad použití programu:

Hledáme slovo -> slovo nebylo nalezeno -> přidání slova -> hledání anglických a českých slov