

## 5. domácí úloha: Lexikografické řazení řetězců

S řadícími algoritmy jste se již setkali nejenom na cvičení z PDV a ALG, ale jistě i ve své vlastní praxi. Úloha řazení se totiž velice často objevuje jako podúloha v různých algoritmech. Pro efektivní řazení je důležité vybrat správný řadící algoritmus. Na cvičení jste měli možnost zrychlit pomocí paralelizace několik algoritmů, které jsou dobré především pro numerické řazení. V rámci této domácí úlohy si zkusíte naimplementovat paralelní verzi řadícího algoritmu Radix sort, který použijete na lexikografické seřazení řetězců stejné délky.

**Vaším úkolem je co nejrychleji lexikograficky seřadit vektor řetězců.** Řetězce mají stejnou délku, jsou velmi krátké a obsahují náhodné znaky z omezené abecedy. Pro takovouto datovou sadu je vhodné použít algoritmus Radix sort. Tento řadící algoritmus již také znáte z předmětu ALG [[https://cw.fel.cvut.cz/wiki/\\_media/courses/a4b33alg/alg09\\_2011.pdf](https://cw.fel.cvut.cz/wiki/_media/courses/a4b33alg/alg09_2011.pdf)].

Jelikož jsou všechny řetězce stejně dlouhé, nemusíte v tomto případě ošetřovat speciální případy, kdy řetězce nemají stejnou velikost. Než se pustíte do samotné implementace, zkuste se nejdříve zamyslet nad tím, která z verzí Radix sortu (popsaných například zde [[https://en.wikipedia.org/wiki/Radix\\_sort](https://en.wikipedia.org/wiki/Radix_sort)]) je pro daná data (krátké řetězce) na paralelizaci nejvhodnější.

Stáhněte si balíček [hw06\\_sort.zip](https://cw.fel.cvut.cz/wiki/_media/courses/b4b36pdv/tutorials/hw06_sort.zip)

[[https://cw.fel.cvut.cz/wiki/\\_media/courses/b4b36pdv/tutorials/hw06\\_sort.zip](https://cw.fel.cvut.cz/wiki/_media/courses/b4b36pdv/tutorials/hw06_sort.zip)]. Vaším úkolem je doimplementovat tělo metody `radix_par` v souboru `sort.cpp`. Tato metoda má za úkol správně seřadit vektor pointerů na řetězce podle lexikografického pořadí řetězců. Doporučujeme si před samotnou implementací projít `sort.h` a `main.cpp`, kde najdete další nápovědu a vysvětlení dílčích částí kódu.

Zazipované soubory `sort.h` a `sort.cpp` odevzdávejte do systému BRUTE do **6. 4. 23:59 CET pro střední cvičení** a **7. 4. 23:59 CET pro čtvrté cvičení**. Před odevzdáním se ujistěte, že jste z kódu odebrali všechny ladící výpisy.

Za úlohu můžete získat maximálně **2 body** v závislosti na rychlosti vašeho řešení.

[courses/b4b36pdv/tutorials/hw\\_05.txt](#) · Last modified: 2023/03/01 13:07 by fiedlda1