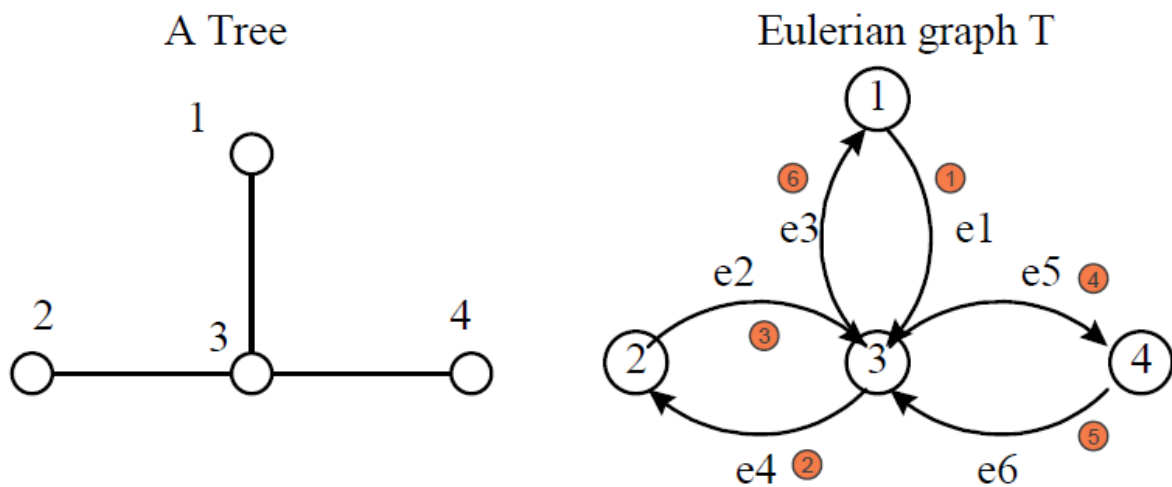


1. Teorie

Pro úspěšné splnění tohoto úkolu bylo zapotřebí správné pochopení a použití dvou důležitých teoretických celků. Jím jsou Eulerova cesta a přiřazení pořadí preorder vrcholům binárního stromu.

Eulerova cesta slouží k vytvoření grafu s orientovanými hranami z binárního stromu. Z binárního stromu $T = (V, E)$, kde V jsou vrcholy a E jsou hrany mezi vrcholy, je vytvořen orientovaný graf $T' = (V, E')$, kde E' je množina orientovaných hran. Každá hrana (u, v) z binárního stromu je nahrazena dvojicí orientovaných hran $\langle u, v \rangle$ a $\langle v, u \rangle$.



Obrázek 1. Tvorba orientovaného grafu

Na základě orientovaného grafu je pak možné vytvořit seznam sousednosti, který slouží k vytvoření Eulerovy cesty v grafu. Následně je potřeba eulerovu cestu „rozříznout“, čímž vznikne seznam. Toho lze docílit tak, že hrana, která by normálně ukazovala do kořenového uzlu bude ukazovat na uzel, ze kterého sama vychází.

Následuje ohodnocení hran v tomto seznamu, což lze dále použít při přiřazování preorder pořadí vrcholům binárního stromu. Hran v tomto seznamu ohodnotíme vahami tak, že pokud se jedná o dopřednou hranu, její váhové ohodnocení je 1, pokud jde o hranu zpětnou, její váhové ohodnocení je 0. Nad tímto polem vah provedeme sumu suffixů, čehož dosáhneme uspořádání dopředných vah. Toto pořadí je ovšem získané průchodem eulerovy cesty od konce, tedy je zapotřebí provést korekci invertováním pořadí ohodnocených vah.

2. Analýza složitosti

- | | |
|------------------------|-------------|
| • Spočtení pole Etour: | $O(c)$ |
| • Ohodnocení hran | $O(\log n)$ |
| • Přiřazení Etour(e) | $O(1)$ |

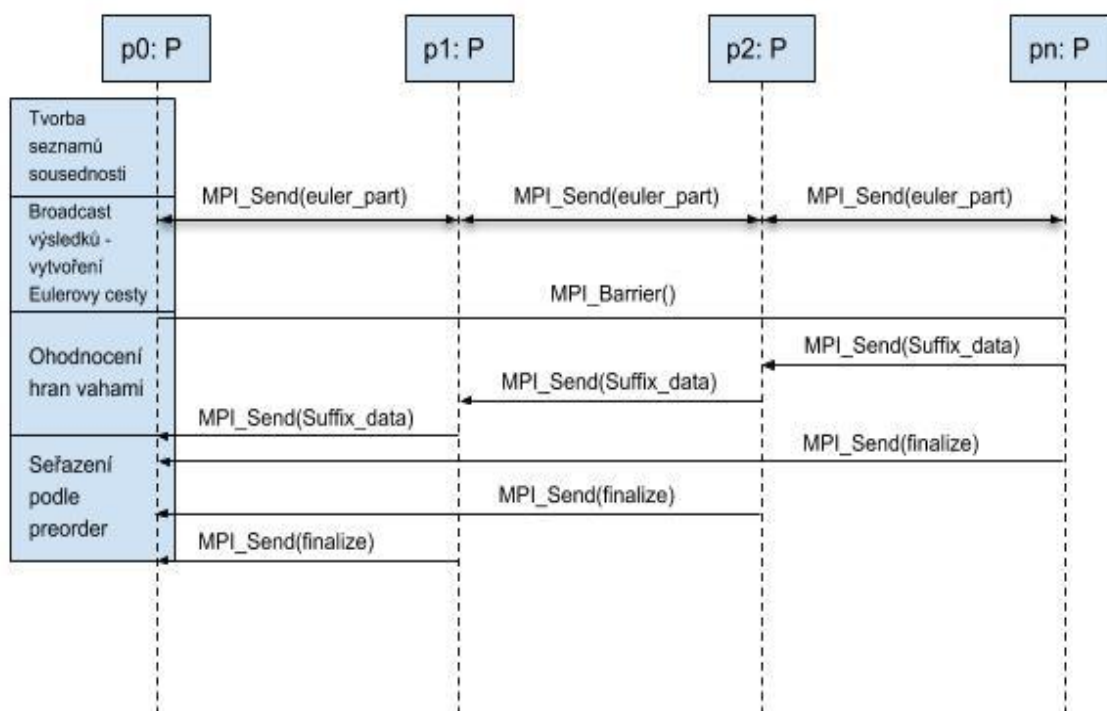
3. Implementace

K implementaci byl použit jazyk c++ a knihovna Open MPI (Message Passing Interface library).

Jako vstup aplikace lze použít řetězec alfanumerických znaků, které symbolizují jednotlivé uzly binárního stromu. Tvorba stromu probíhá tak, že index pozice charakteru ve vstupním řetězci odpovídá indexu uzlu v binárním stromě. Například pro řetězec „ABCD“ vznikne binární strom, kde „A“ je kořen stromu, „B“ a „C“ jsou jeho levé a pravé dítě, a „D“ je jediné dítě uzlu „B“. Počet použitých procesorů je $2n - 2$, kde n značí počet vstupních symbolů (nebo také délku vstupního řetězce). Implementace algoritmu proběhla podle postupu popsání v přednášce 007 předmětu PRL, s výjimkou počítání suffixu, který je implementován s lineární časovou složitostí. Tento problém je znázorněn na grafu v sekci 4.

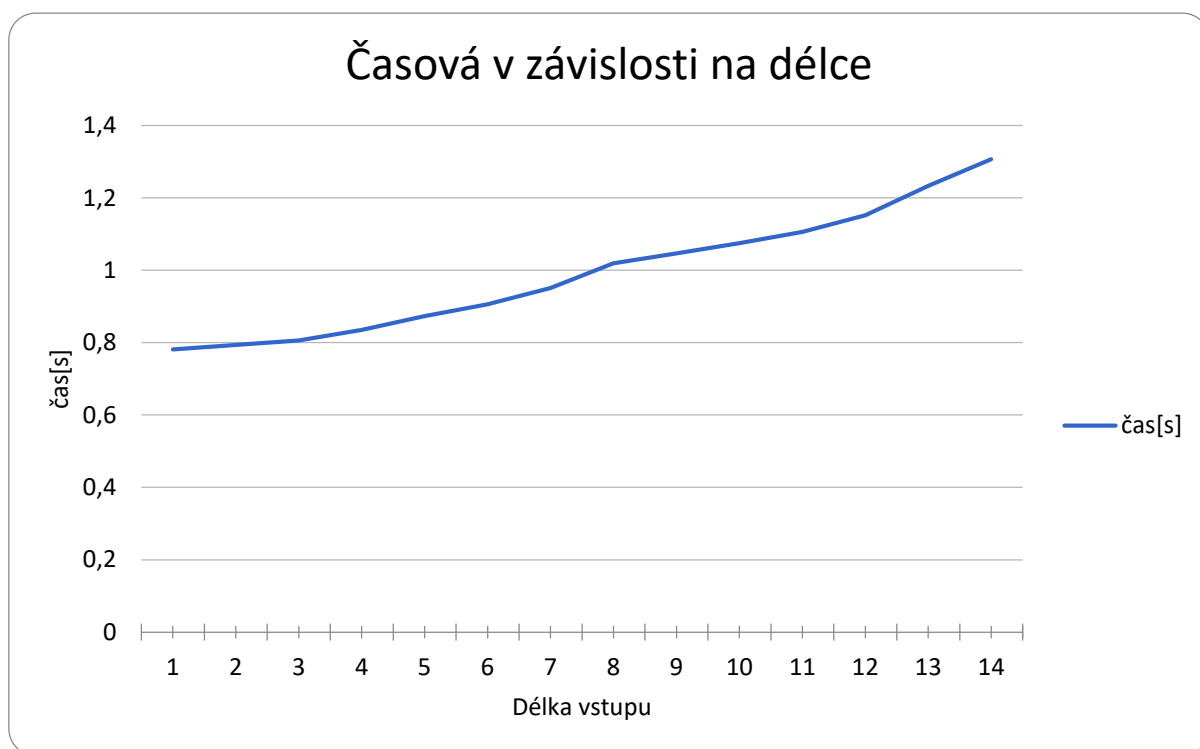
3.1 Sekvenční diagram

Následující diagram znázorňuje komunikaci mezi procesory po spuštění aplikace



4. Experimentování

Během testování aplikace jsem experimentoval s různými délkami vstupních řetězců. Výsledky tohoto testování znázorňuje přiložený graf.



5. Závěr

Během projektu jsem si mohl vyzkoušet implementaci Preorder průchodu binárním stromem pomocí knihovny Open MPI. Také musím konstatovat, že část implementující počítání sufixu má lineární složitost, namísto logaritmické, což je viditelné na grafu v sekci 4.