

Paralelní a distribuované algoritmy

Přiřazení pořadí preorder vrcholům

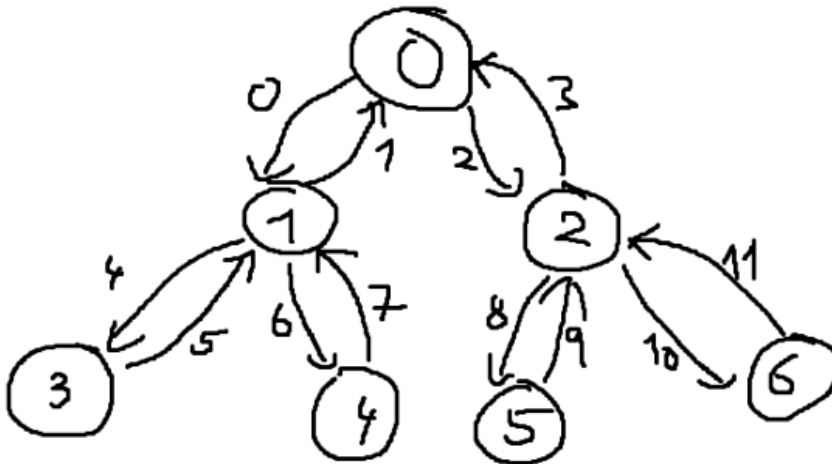
Analýza algoritmu

Algoritmus sa skladá z niekoľkých krokov.

- Spočítanie Eulerovskej cesty
- Inicializácia váh
- Výpočet sumy suffixov
- Korekcie výsledku

Eulerovská cesta

V prvom kroku sú uzly a hrany stromu očíslované indexovaním od jedného a postupným prechodom zhora nadol a zľava doprava. Názorný príklad môžeme vidieť na obrázku.



Tento formát nám umožní matematicky vyjadriť a dopočítať rôzne vzťahy v strome, ako smer hrany, reverzné hrany, rodičovské vrcholy a podobne. Vďaka tomu má každý procesor znalosť len o číslach svojej hrany a počte vstupov/hrán, no vie zistiť svojho nasledovníka v eulerovskej ceste. To zisťuje podľa podmienok stanovených v pseudokóde v slidoch prednášok predmetu PRL. Každý proces zistí svojho nasledovníka a následne sa cez `MPI_Allgather()` rozpošle medzi všetky procesy. Vznikne teda pole, kde na indexe čísla hrany je uložené číslo jeho nasledovníka.

Inicializácia váh

Inicializácia váh prebehne veľmi jednoducho. Váha má byť 1 pre napredné hrany a 0 pre ostatné. Môžeme si všimnúť že napredné hrany budú mať vždy párne číslo. Teda stačí spraviť pole striedajúce hodnoty 1 a 0. Každý procesor vytvorí váhu pre svoju hranu a znova sa všetky rozpošlú cez `MPI_Allgather()`.

Suma Suffixov

Keď už máme zoznamy váh a následníkov, môžeme spočítať sumu suffixov. Iterácia výpočtu sa drží pseudokódu v prednáškach. Následne každý procesor obsluhujúci doprednú hranu vykoná korekciu aby sme dostali opačné poradie a pošle svoju váhu procesoru s číslom 3. Ten pozbiera všetky váhy a zoradí ich. Uzly v takomto poradí sú nasledné vypísané na štandardný výstup.

Časová zložitosť

Teoretická **časová zložitosť** algoritmu sa dá odvodiť takto:

- Výpočet Etour = $O(c)$ (konštantný čas, každé cpu jeden výpočet)
- Inicializácia váh = $O(c)$ (konštantný čas, každé cpu vlastnú váhu)
- Výpočet Sumy suffixov = $O(\log n)$ (logaritmická zložitosť, minimálny počet iterácií)
- Korekcia výsledku = $O(c)$ (konštantný čas, každé cpu vlastnú korekciu)

Celková časová zložitosť algoritmu je teda $O(\log n)$.

Priestorová zložitosť

teda počet procesorov potrebných pre algoritmus je rovný:

$$2 * (n - 1).$$

Teda napríklad pre vstup o dĺžke 7 potrebujeme $2*(7-1) = 12$ procesorov.

Záver

Implementácia algoritmu je funkčná no pri testovaní som objavil bug, kde z neznámeho dôvodu program nefunguje pre vstup o dĺžke 2 a 5 na serveri merlin. Zvyšné vstupy fungujú správne.

```
./test.sh ABCDEFG
ABDECFG
./test.sh ABC
ABC
./test.sh ABCDEFGHIJKL
ABDHIEJKCFLG
```