

Popis

Zeptáme se, zdali můžeme osekát A na B a pak symetricky naopak. Jestliže je B obsažen v A , pak můžeme procházet zároveň (tj. pokud se budeme zanořovat do indexu i , tak musíme být schopni se do něj zanořit v obou stromech, atd..) oba stromy bez toho, aniž bychom při DFS procházce stromu B v A něco přeskočili. Najít ale vhodný počáteční vrchol v A může být problém. Můžeme si tedy DFS projít strom A , ze směrů poskládat slovo (při zanořování připojíme ke slovu číslo hrany, při vynořování zaznačíme v). To stejné provedeme i se stromem B . Nyní máme slova S_A a S_B . Pakliže najdeme v S_A slovo S_B , tak jsme schopni udělat stejnou DFS pochůzku, která je pro B kompletní a pro A minimálně validní (nepřeskakuje žádné vrcholy a je možné vše projít).

Správnost

Rozebraná v sekci *Popis*. Pakliže najdeme výstupní slovo v jiném, pak jsme schopni udělat danou pochůzku bez přeskokování, tedy je možné vybrat kořen, odstranit rodiče a zahodit nějaké množství hran zleva a zprava (tj. prostředek nechat ucelený). Proto jsme schopni větší strom osekát.

Složitost

Každý strom musíme projít a postavit z něj slovo, z toho plyne alespoň $O(a+b)$ paměťově i časově. Vybudování automatu + samotné vyhledávání je lineární k oběma složkám, tedy $O(a+b)$ i s tím, že musíme algoritmus spustit symetricky dvakrát: $out = contains(A, B) \quad || \quad contains(B, A)$.