

### *Příklad (řeka)*

Je dáno povodí řeky ve formě orientovaného stromu, v němž každý vrchol je buď soutok nebo molo (či současně obojí). Pro každý úsek mezi sousedícími vrcholy je dána jejich vzdálenost (malé přirozené číslo). Je dáno malé přirozené číslo  $d$ . Otázka je, zda v povodí existují dvě mola, kde jedno je dosažitelné po proudu od druhého a jejichž vzdálenost je přesně  $d$ . Popište algoritmus, který na danou otázku odpoví.

┌

### *Řešení*

Nechť hloubka je vzdálenost (tj. součet délek úseků řeky na přímé cestě) mola / soutoku od ústí (kořene stromu). V datové struktuře  $S$ , která umí FIND, INSERT, DELETE (hashset, vyhledávací strom, ...), si budeme ukládat tuto hloubku všech vrcholů nad aktuálně řešeným vrcholem. (Hloubky jednotlivých vrcholů si buď předpočítáme, nebo je budeme počítat jednoduše průběžně).

Provedeme DFS, kde v každém vrcholu nejdříve na  $S$  provedeme (samozřejmě jen, pokud je tam molo, jinak pokračujeme do synů) FIND aktuální hloubky mínus  $d$ , a pokud bude úspěšný, tak jsme našli mola, která jsou od sebe po proudu řeky ve vzdálenosti přesně  $d$ . Pokud jsme nenašli, tak pokud je v aktuálním vrcholu molo, provedeme INSERT jeho hloubky, spustíme se na všechny syny a nakonec (pokud je tu molo) provedeme DELETE.

Tím jsou v každém kroku v  $S$  hloubky přesně těch mol, která jsou po proudu od aktuálního vrcholu. Tedy pokud taková 2 mola existují, tak je najdeme. Časová složitost je  $O(n)$  krát součet časových složitostí FIND, INSERT a DELETE, jelikož projdeme každý vrchol a v každém, kde je molo (což můžou být všechny) provedeme tyto 3 operace. Tedy pro hashset je to průměrně  $O(n)$ , pro vyhledávací strom  $O(n \cdot \log(n))$ .

└