

Disjunktní množiny (Union-Find)

Uvažujme implementaci pomocí stromů s heuristikou *union by rank* (ale bez komprese cest). Ukažte posloupnost operací MAKE-SET, UNION a FIND-SET délky m , z nichž n operací je MAKE-SET, takovou, že její časová složitost je $\Omega(m \log n)$.

Disjunktní množiny (Union-Find)

Chceme přidat operaci $\text{PRINT-SET}(x)$, která vypíše všechny prvky v množině, do které patří prvek x . Jakým způsobem upravit implementaci, aby PRINT-SET bylo lineární vůči velikosti množiny a složitosti ostatních operací se nezhoršily?

- ▶ v případě implementace pomocí zřetěžených seznamů (plytkých stromů)
- ▶ v případě implementace pomocí stromů s *union by rank* a kompresí cest

Disjunktní množiny (Union-Find)

Operace UNION se ve skutečnosti skládá ze dvou operací: FIND-SET a LINK. Ukažte, že posloupnost m operací MAKE-SET, LINK a FIND-SET, v níž se všechny operace LINK objevují před první operací FIND-SET, má složitost $\mathcal{O}(m)$.

Proč stejná úvaha nefunguje ve chvíli, kdy se operace můžou libovolně míchat?

Aplikace disjunktčních množin

Offline minimum

Máme předem dānu posloupnost operací $\text{INSERT}(x)$ a EXTRACT-MIN . Vkládané prvky jsou z množiny $\{1, \dots, n\}$ a každý prvek je vložen právě jednou. Úkolem je vypsāt posloupnost prvků, které by postupně vracely operace EXTRACT-MIN .

Příklad: I(4), I(8), E, I(3), E, I(9), I(2), I(6), E, E, E, I(1), I(7), E, I(5).

1. Jaký bude výstup pro výše uvedený příklad?
2. Jak řešit tento problém obecně? S jakou složitostí?