



Fórum studentů MFF UK

Fórum pro všechny studenty matematicko-fyzikální fakulty UK, informatiky, fyziky i matematiky

[Přejít na obsah](#)

[Pokročilé hledání](#)

- [Obsah fóra](#) < [Informatika LS](#) < [Výuka LS 2. ročník](#) < [PRG005 Neprocedurální programování](#)
- [Změnit velikost textu](#)
- [Napsat e-mail](#)
- [Verze pro tisk](#)

- [FAQ](#)
- [Registrovat](#)
- [Přihlásit se](#)

Zkouška 30. 05. 2016 (Dvořák, Hric)

[Odeslat odpověď](#)

Příspěvek: 1 • Stránka 1 z 1

- [Ohlásit tento příspěvek](#)
- [Odpovědět s citací](#)

Zkouška 30. 05. 2016 (Dvořák, Hric)

od JJJ » 30. 5. 2016 20:40

Prolog:

1) Je zadán obecný graf. (reprezentaci si máme zvolit) Máme nad ním spustit DFS, a ke každému vrcholu si poznamenat čas, kdy jsme do něj vstoupili, a kdy jsme z něj vystoupili.

2) Na vstupu dostaneme posloupnost čísel a číslo V. Máme vrátit všechny možné matematické výrazy, které lze z dané posloupnosti postavit pomocí operátorů +, -, *, // a závorek, a jejichž hodnota je V. Výraz musí využít všechna zadaná čísla, a jejich pořadí nesmí měnit. Dále si máme dávat pozor, abychom ve výrazu nedělili nulou.

Haskell:

1) Pro zadané K máme generovat nekonečný uspořádaný seznam K-tic: uspořádání je definováno tak, že nejprve se třídí podle maximálního prvku v daném seznamu. (jakákoliv k-tice, jejíž maximum je menší nebo rovno 2 bude před k-ticí obsahující číslo 3). Když mají dvě k-tice stejné maximum, řadí se lexikograficky.

Příklad pro K=2:

[[0,0],[0,1],[1,0],[1,1],[0,2],[1,2],[2,0],[2,1],[2,2],[0,3],.....]

2) Máme zadáný binární vyhledávací strom (reprezentaci si máme zvolit), a dvě čísla D, H . Máme vrátit BVS, který vznikl ořezáním vstupního stromu tak, aby v něm byly pouze hodnoty X takové, že $D \leq X \leq H$

Velký příklad:

Mějme konečný deterministický automat. (reprezentaci si máme zvolit)

Cílem je najít nejkratší synchronizační slovo s , nebo říct, že neexistuje.

Synchronizační slovo je takové slovo nad vstupní abecedou, že když nad ním spustíme výpočet, tak se automat vždy dostane do nějakého stejného stavu bez ohledu na to, ve kterém stavu začínal.

(https://en.wikipedia.org/wiki/Synchronizing_word)

Součástí zadání je horní odhad délky slova - pokud synchronizační slovo existuje, má délku nejvýše $(n^3 - n)/6$, kde n je počet stavů automatu.

Úloha je NP-úplná, takže je potřeba řešit to hrubou silou s tím, že máme zkusit najít nějakou heuristiku.

Jinak, ověřit, zda řešení existuje lze v polynomiálním čase. Stačí si všimnout, že synchronizační slovo existuje právě tehdy, když pro každou dvojici stavů existuje slovo, po kterém se z obou dostaneme do stejného stavu. Existenci takového slova pro každou dvojici lze vyřešit iterativním generováním všech možných dvojic, které jsou sjednotitelné. (Stav S je vždy sjednotitelný se stavem S . Navíc, pokud stavy A, B jsou sjednotitelné, a existuje písmeno, kterým se dokážu na jeden krok dostat z C do A , a zároveň z D do B , pak i stavy C, D jsou sjednotitelné.) Pokud se mi takto povede vygenerovat všechny dvojice stavů, pak každé dva stavy jsou sjednotitelné, a tedy synchronizační slovo existuje. Pokud se mi takto nějakou dvojici vygenerovat nepodaří, pak dané dva stavy nejsou sjednotitelné, a tedy synchronizační slovo neexistuje. Ale tohle už jsem tam stihl jenom popsat.

JJJ

[Nahoru](#)

[Odeslat odpověď](#)

Příspěvek: 1 • Stránka 1 z 1

[Zpět na PRG005 Neprocedurální programování](#)

Přejít na: PRG005 Neprocedurální programování



Přejít

Kdo je online

Uživatelé procházející toto fórum: Žádní registrovaní uživatelé a 1 návštěvník

- [Obsah fóra](#)
- [Tým](#) • [Smazat všechny cookies z fóra](#) • Všechny časy jsou v UTC + 1 hodina

POWERED BY

Český překlad – phpBB.cz