



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 26.6.2014 (Dvořák + Hric)

[Odeslat odpověď](#)

Příspěvků: 2 • Stránka 1 z 1

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

Zkouška 26.6.2014 (Dvořák + Hric)

od **Twister** » 26. 6. 2014 17:35

Prolog

1. To spamati asi nenapisem, napiste to niekto iny :/

2. Mame dany nedokonaly BVS, mame vypisat dvojice vrcholov, ktore porusuju podmienky BVS. Kazdy vrchol a list ma nejaku hodnotu, co je unikatne cele cislo. Uz to tu par krat bolo.

Datova struktura:

Kód: [Vybrat vše](#)

```
data BVS a = N nil | NT (BVS a) a (BVS a)
```

(alebo nieco podobne)

Haskell

3. HTML serializer. Mame dany obecny strom, ktory ma vo vrcholoch a listoch nazvy HTML tagov, napr. "html", "body", "a", atd. Cielom bolo:

a) napisat typ konštruktoru tej datovej struktury

b) vypisat tagy do stringu podľa DFS priechodu stromom zľava - teda vlastne tak, aby to bolo validne HTML.

Signatura:

Kód: [Vybrat vše](#)

```
vypis (Num a, Ord a):: NTree a -> String
```

Příklad:

Kód: [Vybrat vše](#)

```
Strom:      html
            /  \
          head  body
             /   \
            a     h2
```

```
Vypis: <html><head></head><body><a></a><h2></h2></body></html>
```

4. Mame body v niekoľkodimenziálnom priestore. Vzdialenosť medzi bodmi je určená pomocou Manhattanskej metriky. Funkcia dostane zadanú dvojicu bodov (prihradok), a zoznam ďalších bodov. Singatúra funkcie bola:

Kód: [Vybrat vše](#)

```
zarad :: (Num a, Ord a) => ([a], [a]) -> [[a]] -> ([[a]], [[a]])
```

Kazdy bod zo zoznamu bolo treba zaradiť do tej bližšej z tých 2 prihradok, teda do tej, ktorá je bližšie podľa Manhattanskej metriky.

Napríklad:

Kód: [Vybrat vše](#)

```
> zarad ([0,0] [5,5]) [[1,2], [6,4], [0,-1]]
([[0,0], [1,2], [0,-1]], [[5,5], [6,4]])
```

Big One

Je daný počet strojov n , a zoznam výrobkov (každý výrobok má dané číslo, dobu koľko trvá jeho výroba, čas odkedy je možné ho vyrobiť, a čas dokiaľ musí jeho výroba skončiť). Môžeme predpokladať, že doba výroby sa vojde do tohoto intervalu od-do (teda každý výrobok je vyrobiteľný). Stroje sú navzájom zameniteľné, výrobky tiež. Každý stroj môže samozrejme naraz vyrábať len jeden výrobok.

Ulohou bolo vrátiť rozvrh výroby (teda zoznam zaznamov typu [číslo výrobku, čas začatia výroby, číslo stroja]), tak aby sa maximalizoval počet vyrobených výrobkov. Nemusia sa dať vyrobiť všetky. Málo sa to riešilo s heuristikou. Dvořák nám poradil, aby sme to písali zhora, teda pomocne funkcie až nakoniec, pre prípad, že by sme to nestihli.

Uloha bola veľmi nečakane NP úplná 😊

Podľa mňa bol na to oveľa lepší Prolog, ale viem, že to pár ľudí robilo aj v Haskell.

Ja som to robila tak, že som sa najprv pokúsila najst rozvrh, ktorý vyrobí všetky výrobky (teda m výrobkov), ak sa to nepodari, skúsi vyrobiť $m-1$... atď., až po skúsi vyrobiť jeden výrobok. Ak sa mu nepodari ani jeden, zlyhá.

Rozvrh som sa snažila vyrobiť pomocou predikatov `between` a kontrolovania podmienok.

Bolo treba splniť podmienky:

- jeden stroj nikdy nevyrába viac vecí naraz
- výroba každého výrobku prebieha v jeho povolenom intervale

Este neviem, ci moje riesenie bolo aj spravne, na ustnu cast idem az zajtra. Ale prislo mi to ako jedna z lahsich uloh.

Twister

[Nahoru](#)

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

[Re: Zkouška 26.6.2014 \(Dvořák + Hric\)](#)

Od [jira](#) » 26. 6. 2014 18:45

Twister píše: **Prolog**

1. To spamati asi nenapisem, napiste to niekto iny :/

Byl dán seznam objektů, kde každý objekt obsahuje dvojice klíč-hodnota. Úkolem bylo pro každý klíč vypsat jakých hodnot nabývá.

Pokud nějaký objekt klíč neobsahuje, tak je hodnota undefined. Objekty jsou prostě seznamy.

Kód: [Vybrat vše](#)

```
[
  [ jmeno-"xy", vek-30, vaha-90],
  [ jmeno-"xyz", vek-35, vaha-80]
  [ jmeno-"ab", vaha-80]
]

[ jmeno-["xy", "xyz", "ab"], vek-[30, 35, undefined], vaha-[90, 80, 80]]
```

Ten velký příklad, jsem dělal tak, že jsem přednostně plánoval úlohy s nejkratší dobou zhotovení a vždy když byl nějaký výrobek hotov, tak jsem naplánoval další.

Bylo mi řečeno, že heuristika je trochu moc jednoduchá a dostal jsem za dvě.

[jira](#)

Matfyz(ák|ačka) level I

Příspěvky: 11

Registrován: 29. 6. 2011 09:43

Typ studia: Informatika Bc.

[Nahoru](#)

Zobrazit příspěvky za předchozí: Všechny příspěvky ▼ Seřadit podle Čas odeslání ▼ Vzestupně ▼

[Přejít](#)

[Odeslat odpověď](#)

Příspěvků: 2 • 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](#)