Midterm, 23.11.2015 Meno,priezvisko:

Pravidlá:

- **Ľas 90 minút**
- **môžete používať akékoľvek ale len Vami prinesené materialy,**
- **knihy, poznámky, listingy, é**
- **nie je možné používať iné elektronické,**
- **nie je možné zdieľať akékoľvek pomôcky a materialy,**
- **kolektívne riešenia sa netolerujú,**
- **každý príklad riešte na papieri obsahujúcom jeho zadanie,**
- **midterm obsahuje 4 príklady spolu za 25 bodov.**

1. **[6 bodov – Maticový]** Obdĺžniková matica je zoznam rovnako dlhých riadkov celých ľísel a je definovaná ako **`type Matica=[[Int]].`** Jemne inšpirovaní *piškvorkami*, definujte funkcie, ktoré zisťujú prítomnosť *piškvorky* v riadku, stĺpci a v oboch diagonálych smeroch. Ak sa zamyslíte nad tromi podobnými úlohami, zistíte, že sú inštanciou jednej, ktorú ak urobíte, tak máte 3-in-1.
   - **[2body]** Definujte funkciu **`patVRiadku::Matica->Bool`**, ktorá zistí, ľi v matici existuje riadok obsahujúci päťnúl vedľa seba.
   - **[2body]** Definujte funkciu **`patVStlpci::Matica->Bool`**, ktorá zistí, ľi v matici existuje stĺpec obsahujúci päťnúl pod sebou. Riešenie transponujúce maticu dostane max. 1.5 bodu.
   - **[2body]** Definujte funkciu **`patNaDiagonale::Matica->Bool`**, ktorá zistí, ľi sa v matici nachádza päťnúl vedľa seba v niektorom šikmom diagonálnom smere. Bez obáv a testov predpokladajte, že na vstupe je skutoľne obdĺžniková matica s rovnako dlhými riadkami.

```
Príklad: m = [[1,0,0,1,0,0,1,0,0],
              [1,0,0,1,0,0,1,0,0],
              [1,0,0,1,0,0,0,0,0],
              [1,0,0,1,0,0,1,0,0],
              [1,0,1,1,0,0,1,0,1]]
patVRiadku m = patVStlpci m = patNaDiagonale m = True
```

-------- v prípade potreby riešte túto úlohu na zadnú stranu **TOHOTO listu**----------

2. [8 bodov – foldový]
V tomto zadaní je **/** funkcia deleno, teda **(\x->\y->x/y)**, a teda **6/2==3.0** a **2/6==0.33333.**

- **[1 bod]** vypoľítajte výrazy (jeden je **foldr** a druhý **foldl**).
  foldr (/) 3 [3,3,3] =
  foldl (/) 3 [3,3,3] =
- **[1 bod]**
  Ktorý z výrazov **foldl (/) 3 [1,2,3], foldr (/) 3 [1,2,3]** je väľší/menší/rovnaké.

V každej ďalšej časti úlohy si stačí vybrať jednu z verzií foldr alebo foldl, podľa preferencií.

- **[2body]** do definície funkcie **vsetkyRovnake::(Eq t)=>[t]->Bool** doplňte výrazy za symboly **?**, aby funkcia vrátila **True**, ak sú všetky prvky vstupného zoznamu rovnaké, inak **False**.
  vsetkyRovnake xs = ? foldr ? ? xs, vsetkyRovnake xs = ? foldl ? ? xs.

- **[2body]** do definície funkcie **identicke::(Eq t)=>[t]->[t]->Bool** doplňte výrazy za symboly **?**, aby funkcia vrátila **True**, ak sú oba zoznamy rovnaké, inak **False**. Pochopiteľne, porovnávanie zoznamov nie je povolené, môžete porovnávať len hodnoty typu **t**.
  identicke xs ys = ? foldr ? ? xs, identicke xs ys = ? foldl ? ? xs.

- **[2body]** do definície funkcie **podmnozina::(Eq t)=>[t]->[t]->Bool** doplňte výrazy za symboly **?**, aby funkcia vrátila **True**, ak všetky prvky prvého zoznamu sa nachádzajú v druhom, inak **False**. Riešenie **neobsajuhúce** funkciu elem**::(Eq t)=>[t]->[t]->Bool** dostane +0.5 bodu.
  podmnozina xs ys = ? foldr ? ? xs, podmnozina xs ys = ? foldl ? ? xs.

-------- v prípade potreby riešte túto úlohu na zadnú stranu **TOHOTO listu**----------

3. [7 bodov – BVS]

Toto je definícia parametrického binárneho stromu s hodnotami typu **t** vo vrcholoch.

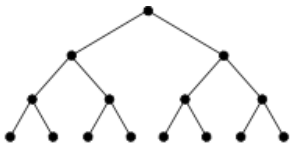
```
data BVS t = Node (BVS t) t (BVS t) | Nil deriving (Show, Eq, Ord)
```

Vrchol **Node left value right** voláme list, **ak left=right=Nil**, inak je to vnútorný vrchol. List je teda len **Node Nil val Nil.** Ak vieme hodnoty typu **t** porovnávať, tak má zmysel sa baviť i o tomto strome spĺňajúcom vlastnosti binárneho vyhľadávacieho stromu. V príklade predpokladáme, že vstupy do definovaných funkcií spĺňajú túto sémantickú vlastnosť, preto ju neoverujte a definujte:

- **[2 body]** funkcie **vyska, velkost::(BVS t)->Int**, ktoré vrátia výšku a počet vrcholov stromu. Výška aj veľkosť stromu **Nil** sú 0**.**
- **[1 bod]** funkciu uplny**::(BVS t)->Bool**, ktorá vráti True, ak strom je úplný binárny strom, t.j. výška ľavého a pravého podstromu **v každom vrchole** je rovnaká. Za efektívne riešenie +1 bod.
- **[2 body]** funkciu vrcholy**::(BVS t)->([t],[t])**, ktorá vráti dvojicu zoznamov hodnôt vnútorných vrcholov a listov. Na poradí hodnôt v zoznamoch nezáleží. Za riešenie, ktoré nepoužíva ++/append +0.5 bodu.
- **[2 body]** funkciu **kty::Int->(BVS t)->t**, ktorá vráti k-ty najmenší prvok stromu, ak taký existuje. Prípad, že neexistuje môže skončiť chybou, teda neriešte. Prvý prvok má index 0.

Príklad:

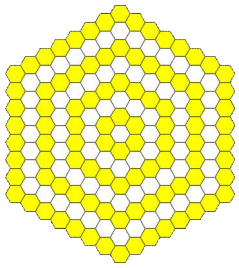
```
t = Node (Node (Node Nil 1 Nil) 3 (Node Nil 4 Nil)) 5 (Node (Node Nil 6 Nil) 7 (Node Nil 9 Nil))
vyska t = 3
velkost t = 7
uplny t = True
vrcholy t = ([3,5,7],[1,4,6,9])
kty 0 t = 1
kty 1 t = 3
kty 2 t = 4
kty 3 t = 5
```



úplný binárny strom

-------- v prípade potreby riešte túto úlohu na zadnú stranu **TOHOTO listu**----------

4) **[4 body] ï ġesŠuholníkový**
a) [2body] Na obrázku vidíte ġesŠuholníkový vļ elí plást. Poļ ty malých ġesŠuholníļ kov v plástoch s rastúcim rozmerom tvoria postupnosŠ1,7,19, 37 ... ZávislosŠmusíte objaviŠ.. Definujte nekoneļ ný zoznam ġesŠuholníkových ļ ísel **`sc::[Int]`**, ktorého zaļ iatok má tvar `[1,7,19,37, …].`
b) [2body] Definujte funkciu **`jeSC::Int->Bool`**, ktorá zistí, ļ i ļ íslo je ġesŠuholníkové ļ íslo, príklad: **`jeSC 61==True,jeSC 52==False.`**



-------- v prípade potreby rieġte túto úlohu na zadnú stranu **TOHOTO listu**----------