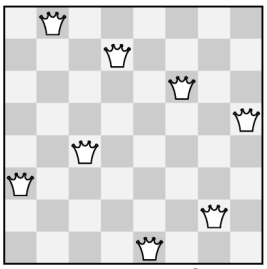
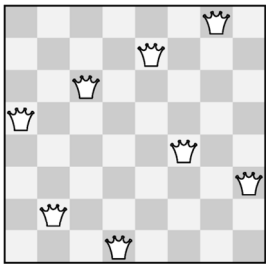
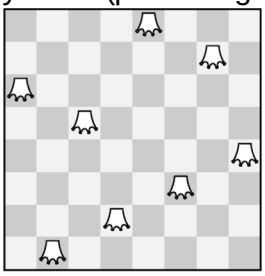
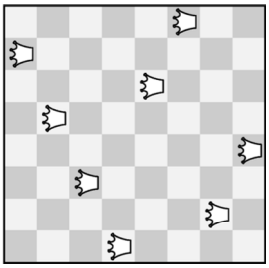
Midterm, 22.11.2016 Meno,priezvisko:

Pravidlá:

- **Ļas 90 minút**
- **môġete pouġívaŠakékoĎvek ale len Vami prinesené materialy,**
- **knihy, poznámky, listingy, é**
- **nie je moġné pouġívaŠniĻ elektronické,**
- **nie je moġné zdieĽaŠakékoĎvek pomôcky a materialy,**
- **kolektívne rieġenia sa netolerujú,**
- **kaġdý príklad rieġte na papieri obsahujúcom jeho zadanie,**
- **midterm obsahuje 4 príklady spolu za 25+1 bodov.**

1.[6 bodov - Dámy]

Tento príklad môġete rieġiŠv Haskelli, Go alebo Pythone (podĽa váġho výberu). Zadanie je v HS.

Rieġenie problému 8-dám reprezentjeme ako permutáciu riadkov, v ktorých sa jednotlivé dámy nachádzajú, teda permutáciu Ļísel 0..7. V tomto príklade budete zisŠovaŤ ktoré rieġenia sú **symetricky rovnaké**. Vġetko ilustrujeme na zobrazenom rieġení `[5,0,4,1,7,2,6,3]` (1.obr.). Následujúce obrázky 2, 3, 4 potom zobrazujú symetriu horizontálnu, vertikálnu a diagonálnu.
V následujúcich funkciách dostanete rieġenie problému N-dám, kde N nie je konġtanta 8.

- **[2 body]** Definujte funkcie symHoriz, symVertik :: [Int] -> [Int], ktoré vrátia horizontálne a vertikálne symetrické rieġenie, teda `[3,6,2,7,1,4,0,5]` a `[2,7,3,6,0,5,1,4]`.

- **[2 body]** Definujte funkciu symDiag :: [Int] -> [Int], ktorá vráti diagonálne symetrické rieġenie, teda `[1,3,5,7,2,0,6,4]`. Ktorú z dvoch uhlopriečok si vyberiete je na vás.

- **[1 bod]** Definujte funkcie sym90, sym180 a sym270 :: [Int] -> [Int], ktoré vrátia rieġenie otoĻ ené o 90, 180 a 270 stupŇov v niektorom smere, teda `[4,6,0,2,7,5,3,1]`, `[4,1,5,0,6,3,7,2]`, `[6,4,2,0,5,7,1,3]`, v naġom príklade je to v smere hodinových ruĻ iĻ iek.
  Hint: tu uģ moc neprogramujte, skúste to vyskladaŠz funkcií z prvých dvoch bodov úlohy, heslo k úspechu je skladanie symetrií.

- **[1 bod]** Definujte funkcie symetrické :: [Int] -> [Int] -> Bool, ktoré o dvoch rieġeniach problému N dám rozhodne, Ļi sú nejako symetrické, pouģite to, Ļo ste doposiaĎvytvorili.

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

2. [8 bodov ï foldový]
V tomto zadaní je funkcia **<** na type **Bool**, teda **False < True** je **True,** inak **False.**

- **[1 bod]** vypoļítajte výrazy (jeden je **foldr** a druhý **foldl**).
  foldr (<) True [False, False]
  foldl (<) False [True, True]

- **[1 bod]**
  Ktorý z výrazov **foldr (<) False [False, False], foldl (<) True [True, True]** je väļ ší/menší/rovnaké.

V každej Ņalšej ļ asti úlohy si staļ í vybraŠjednu z verzií foldr alebo foldl, podĎa preferencií.

- **[2body]** Do definície funkcie **nieVsetkyRovnake::(Eq t)=>[t]->Bool** doplŕte výrazy za symboly **?**, aby funkcia vrátila **True**, ak obsahuje aspoŕ dva rôzne prvky, inak **False**.
  nieVsetkyRovnake xs = foldr ? ? xs,
  nieVsetkyRovnake xs = foldl ? ? xs.

- **[2body]** Do definície funkcie **disjunktne::(Eq t)=>[t]->[t]->Bool** doplŕte výrazy za symboly **?**, aby funkcia vrátila **True**, ak oba zoznamy nemajú sploļ ný prvok, inak **False**. PochopiteĎne, porovnávanie zoznamov nie je povolené, môžete porovnávaŠlen hodnoty typu **t**.
  disjunktne xs ys = ? foldr ? ? xs, disjunktne xs ys = ? foldl ? ? xs.

- **[2body]** Do definície funkcie **prienik::(Eq t)=>[t]->[t]->[t]** doplŕte výrazy za symboly **?**, aby funkcia vrátila prienik zoznamov, teda prvky nachádzajúce sa v oboch, na poradí prvkov nezáleží. Prvky vo vstupných zoznamoch tvoria množinu, teda v rámci zoznamu sa neopakujú. Rieženie **neobsajuhúce** funkciu elem**::(Eq t)=>[t]->[t]->Bool** dostane **+0.5 bodu**.
  prienik xs ys = ? foldr ? ? xs, prienik **xs ys = ? foldl ? ? xs.**

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

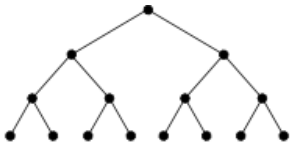
[6 bodov – BVS]
Toto je definícia parametrického binárneho stromu s hodnotami typu **t** vo vrcholoch.
data BVS t = Node (BVS t) t (BVS t) | Nil deriving (Show, Eq, Ord)
Vrchol **Node left value right** voláme list, **ak left=right=Nil**, inak je to vnútorný vrchol. List je teda len **Node Nil val Nil.** Ak vieme hodnoty typu **t** porovnávať, tak má zmysel sa baviť i o tomto strome spĺňa vlastnosti binárneho vyhľadávacieho stromu. V príklade predpokladáme, že vstupy do definovaných funkcií **spĺňajú** túto sémantickú vlastnosť, preto ju neoverujte a definujte:

- **[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 **fromTo::(Ord t)=>t->t->(BVS t)->[t]**, ktorá pre volanie **fromTo a b tree** vráti usporiadaný zoznam prvkov binárneho vyhľadávacieho stromu väčších ako **a** a menších ako **b**. Myslite na efektívnosť aj v prípade, ak je strom veľký a interval (**a;b**) malý.

- **[3 body]** funkciu **jeVyvazeny::(BVS t)->Bool**, ktorá zistí, či binárny strom je vyvážený, t.j. výšky oboch podstromov ľubovoľného vrchola sa líšia najviac o 1.
Príklad:

```
t = Node (Node (Node Nil 1 Nil) 3 (Node Nil 4 Nil)) 5 (Node (Node Nil 6 Nil) 7 (Node Nil 9 Nil))
uplny t = True
fromTo 2 7 t = [3,4,5,6]
jeVyvazeny t = True
```


úplný binárny strom

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

4) **[6 bodov] Postupnosti**

Príklad je o nekonečných celočíselných postupnostiach s typom **`type Postupnost= [Int].`**

- **[1bod]** Definujte **`aritmeticka prvy delta::Int->Int->Postupnost`**, ktorej hodnotami sú prvky aritmetickej postupnosti s prvým členom a deltou, príklad:
  `take 10 (aritmeticka 1 3) = [1,4,7,10,13,16,19,22,25,28]`
- **[1bod]** Definujte **`geometricka prvy quocient::Int->Int->Postupnost`**, ktorej hodnotami sú prvky geometrickej postupnosti s prvým členom a quocientom, príklad:
  `take 10 (geometricka 1 (-2)) = [1,-2,4,-8,16,-32,64,-128,256,-512]`
- **[1bod]** Definujte **`sucet p1 p2::Postupnost->Postupnost->Postupnost`**, ktorá vráti súcet dvoch postupností, príklad:
  `take 10 (sucet (aritmeticka 1 3) (geometricka 1 (-2))) = [2,2,11,2,29,-16,83,-106,281,-484]`
- **[2body]** Definujte
  `vseobecna cleny (c,koeficienty)::[Int]->(Int,[Int])->Postupnost,`
  ktorá pre **`vseobecna [`**a_1**`, `**a_2**`… `**a_n**`] (`**c_0**`,[`**c_1**`, `**c_2**`… `**c_n**`])`** vráti nekonečnú postupnosť, ktorej prvé členy sú $a_1, a_2, \ldots, a_n$, a rekurzívny vzťah pre každý ďalší **`je`**
  $a_{n+1} = c_0+c_1a_1+c_2a_2+c_na_n$.
  Predpokladajte, že dĺžky zoznamov iniciálnych členov a koeficientov sú rovnaké, príklad:
  `take 10 (vseobecna [1,0,-1] (1,[1,2,3])) = [1,0,-1,-1,-4,-14,-50,-181,-656,-2379],` lebo
  štvrtý člen: **`1+ 1*1 + 0*2 + -1*3 = -1`**
  piaty člen: **`1+ 0*1 + -1*2 + -1*3 = -4`**
- **[1bod]** Definujte **`aritmeticka prvy delta`** a **`geometricka prvy quocient`** pomocou **`vseobecna.`**

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