

Deklaratív Programozás gyakorlat
Prolog programozás: listakezelés, gráfok
2016.10.20.

Az alábbi feladatok megoldásában, ha másként nem mondjuk, használhat segédeljárásokat, de ezekhez mindig adjon meg fejkommentet. Megoldásában mindig felhasználhatja az előző feladatokhoz megírt eljárásokat.

1. Beszúrás rendezett listába

```
% insert_ord(+RL0, +Elem, ?RL): Az RL szigorúan monoton növő számlista  
% úgy áll elő, hogy az RL0 szigorúan növő számlistába beszúrjuk az Elem  
% számot, feltéve hogy Elem nem eleme az RL0 listának; egyébként RL = RL0.
```

```
| ?- insert_ord([1,3,5,8], 6, L).  
L = [1,3,5,6,8] ? ; no  
| ?- insert_ord([1,3,5,8], 3, L).  
L = [1,3,5,8] ? ; no
```

Használjon feltételes szerkezetet!

A 'graph' adatstruktúrát a következő Mercury-szerű típusdefiníciókkal definiáljuk:

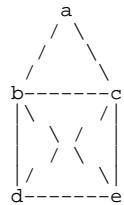
```
% :- type graph == list(edge).  
% :- type edge ---> node-node.  
% :- type node == atom.
```

Eszerint egy Prolog kifejezés a 'graph' típusba tartozik, ha X-Y alakú struktúrák listája, ahol X és Y névkonstansok (atomok).

Az [a1-b1,a2-b2,...,an-bn] 'graph' típusú kifejezés azt az irányítatlan gráfot írja le, amelynek csomópont-halmaza {a1,..,an,b1,..,bn}, és egy (irányítatlan) él vezet ai és bi között, minden i=1,..,n esetén. (Megjegyzés: az így megadott gráfoknak nyilván nem lehet izolált pontja.)

Például az [a-b,a-c], [a-c,b-a], [b-a,a-c], [c-a,a-b] stb. mind ugyanazt a (matematikai értelemben vett) irányítatlan gráfot írják le.

Az [a-b,a-c,b-c,b-d,b-e,c-d,c-e,d-e] kifejezés az alábbi gráfot írja le:



Gyerekkorukban találkozhattak azzal a feladattal, hogy ezt a gráfot egy folytonos vonallal rajzolják meg.

Egy 'graph' típusú [a1-b1,a2-b2,...,an-bn] Prolog listát folytonos vonalnak hívunk, ha b1=a2, b2=a3, ..., b(n-1) = an.

2. Írja meg az alábbi fejkommentnek megfelelő draw/2 Prolog eljárást

```
% draw(+G, -L): Az L folytonos vonal "megrajzolja" a G gráfot, azaz az  
% L folytonos vonal ugyanazt a matematikai értelemben vett gráfot írja  
% le, mint a G Prolog kifejezés.
```

Tehát a "| ?- draw(G, L)." hívás, ahol G adott és L egy változó, felsorolja L-ben az összes olyan folytonos vonalat, amely "megrajzolja" G-t.

Törekedjék minél egyszerűbb megoldásra, nem kell a hatékonysággal foglalkoznia. Használhatja a lists könyvtár eljárásait.

```
| ?- draw([a-b,a-c], L).  
L = [b-a,a-c] ? ;  
L = [c-a,a-b] ? ;  
no  
| ?- draw([a-b,a-c,b-c,b-d,b-e,c-d,c-e,d-e], L), L = [d-e|_].  
L = [d-e,e-b,b-a,a-c,c-b,b-d,d-c,c-e] ? ;  
L = [d-e,e-b,b-a,a-c,c-d,d-b,b-c,c-e] ? ;  
L = [d-e,e-b,b-c,c-a,a-b,b-d,d-c,c-e] ? ;  
L = [d-e,e-b,b-c,c-d,d-b,b-a,a-c,c-e] ? ;  
L = [d-e,e-b,b-d,d-c,c-a,a-b,b-c,c-e] ? ;  
L = [d-e,e-b,b-d,d-c,c-b,b-a,a-c,c-e] ? ;  
L = [d-e,e-c,c-a,a-b,b-c,c-d,d-b,b-e] ? ;  
L = [d-e,e-c,c-a,a-b,b-d,d-c,c-b,b-e] ? ;  
L = [d-e,e-c,c-b,b-a,a-c,c-d,d-b,b-e] ? ;  
L = [d-e,e-c,c-b,b-d,d-c,c-a,a-b,b-e] ? ;  
L = [d-e,e-c,c-d,d-b,b-a,a-c,c-b,b-e] ? ;  
L = [d-e,e-c,c-d,d-b,b-c,c-a,a-b,b-e] ? ;  
no
```

3. (Otthoni feladat)

Írjon Prolog eljárást egy (irányítatlan) gráf fokszámlistájának előállítására. A fokszámlista típusa:

```
% :- type degrees == list(node_degree).  
% :- type node_degree --> node - degree.  
% :- type degree == int.
```

A fokszámlista tehát egy olyan lista, amelynek elemei Cs-N alakú párok, ahol Cs a gráf egy csomópontja, és N a Cs csomópont fokszáma. A csomópontok tetszőleges sorrendben szerepelhetnek a fokszámlistában, de mindegyik pontosan egyszer.

% degree_list(G, Ds): A G gráf fokszámlistája Ds.

```
| ?- degree_list([b-a,a-c], Ds).  
Ds = [b-1,a-2,c-1] ? ; no
```

4. (szorgalmi feladat)

Írjon idraw/2 néven egy Prolog eljárást, amelynek jelentése azonos a 2. feladatban szereplő draw/2 eljárással! Törekedjék minél hatékonyabb megoldásra! Használhatja a ugraphs könyvtárat.

Platónak hívunk egy olyan legalább kételemű listát, amely csupa azonos elemből áll. Az mondjuk, hogy MP egy L listában található maximális plató, ha

- MP az L folytonos része (azaz MP előáll úgy, hogy L elejéről és végéről 0 vagy több elemet elhagyunk),
- MP egy plató és
- MP maximális L-ben, azaz nem lehet sem balra sem jobbra a benne levőkkel azonos, közvetlenül szomszédos elemekkel kiterjeszteni.

Például az L = [a,b,a,c,c,c,b,b] listában két maximális plató van, a 4. pozíción kezdődő [c,c,c] és a 7. pozíción kezdődő [b,b] lista (a listaelemeket 1-től számozzuk).

5. Írjon Prolog eljárást amely a bemenetként kezdődő listáról eldönti, hogy egy platóval kezdődik-e, és ha igen, visszaadja a maximális plató hosszát és az ezután (maradék) elemek listáját.

% pl_kezdetu(+L, ?H, ?M): Az atomokból álló L lista egy H hosszúságú % maximális platóval kezdődik, amelyet az M maradéklista követ.

```
| ?- pl_kezdetu([a,b,a,c,c,c,b,b], H, M).  
no  
| ?- pl_kezdetu([c,c,c,b,b], H, M).  
H = 3, M = [b,b] ? ; no  
| ?- pl_kezdetu([b,b], H, M).  
H = 2, M = [] ? ; no  
| ?- pl_kezdetu([b], H, M).  
no  
| ?- pl_kezdetu([], H, M).  
no  
| ?-
```

6. Írjon olyan Prolog eljárást, amely felsorolja atomok egy adott listájában található maximális platókat, megadva a plató hosszát és az ismétlődő elemet.

% plato(L, H, X): Az L listában található egy H hosszúságú, % X elemekből képzett maximális plató.

```
| ?- plato([a,b,b,b,b,a,a,c,b,b], H, X).  
H = 4, X = b ? ;  
H = 2, X = a ? ;  
H = 2, X = b ? ;  
no
```

7. (szorgalmi feladat)

Az előző feladat kiterjesztéseként írjon olyan Prolog eljárást, amely felsorolja atomok egy adott listájában található maximális platókat, megadva a plató kezdőindexét (1-től számozva), hosszát és az ismétlődő elemet.

% plato(L, I, H, X): Az L listában az I-edik elemtől kezdődően % egy X elemekből képzett, H hosszúságú maximális plató található.

```
| ?- plato([a,b,b,b,b,a,a,c,b,b], I, H, X).  
I = 2, H = 4, X = b ? ;  
I = 6, H = 2, X = a ? ;  
I = 9, H = 2, X = b ? ;  
no
```