

Házi feladat (Modellellenőrzés)

A feladat címe: **Gráfbejáró algoritmus**

Konzulense: **Soltész Adrián**

A modellezendő probléma

Az algoritmuselméletből ismert mélységi bejárás (DFS) iteratív pszeudokódja a következő (n a gráf csúcsainak száma, G a szomszédossági mátrixa, v pedig az algoritmus kiindulásául szolgáló csúcs):

```
DFS( $n, G, v$ ):  
  S.push( $v$ )  
  while S is not empty  
     $v = S.pop()$   
    if (!visited[ $v$ ])  
      visited[ $v$ ] = true  
      for  $w = 0$  to  $n - 1$  do  
        if ( $G[v][w]$  && !visited[ $w$ ])  
          S.push( $w$ )
```

Az S a pszeudokódban egy verem (last in, first out) adatszerkezetet jelöl. Az algoritmus soraiban egy-egy utasítás vagy vizsgálat egy-egy oszthatatlan műveletet jelöl.

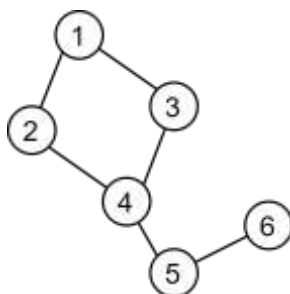
A feladata, hogy implementálja UPPAAL-ban a fent szereplő iteratív mélységi bejárást, és eldöntse, hogy valóban helyes-e a megadott pszeudokód. Az algoritmusnak tetszőleges, a G szomszédossági mátrixával megadott gráfon futnia kell.

A feladat az algoritmust a fenti pszeudokód lépései szerint automataként modellezve várja el, nem pedig teljes egészében függvényként kódolva!

Tipp: A használt adatszerkezetek feldolgozásához viszont érdemes rövid függvényeket írni.

Az ellenőrzendő követelmények

Adott az ábrán látható irányítatlan gráf:



Ezen gráf A szomszédossági mátrixa (ahol $A[i,j] = 1$ jelentése, hogy az i . és j . csúcs közt fut él):

0	1	1	0	0	0
1	0	0	1	0	0
1	0	0	1	0	0
0	1	1	0	1	0
0	0	0	1	0	1
0	0	0	0	1	0

Temporális logikai kifejezések és modellellenőrzés segítségével igazolja az alábbi követelmények teljesülését (illetve a követelmények nem teljesülése esetén ellenpélda segítségével magyarázza meg, miért nem teljesül az adott követelmény)!

1. Igazolja, hogy nincs holtponthoz vezető út a rendszerben.
2. Ellenőrizze, hogy az 1. csúcsból indítva a gráfbejáró algoritmus felderít-e minden csúcsot!
3. Változik-e az algoritmus eredménye, ha törli az ötödik csúcsból a hatodikba vezető élet?
4. Alakítsa át a gráfot egy tetszőleges irányított gráffá, és dokumentálja, hogy a változtatások miben befolyásolták az algoritmus eredményét!
5. Opcionális feladat (valamelyik fenti feladat helyett választható): A megoldást alakítsa át szélességi bejárassá is!