

MDM Lista 12

Weronika Jakimowicz

ZAD. 1.

```
1 odwiedzone = [0] * n
2 skoki = [0] * n
3
4 nozyczki = -1
5
6 def dfsik(j, studnia):
7     odwiedzone[j] = 1
8     skoki[j] = studnia
9     zmienna = 0
10
11     for i in sasiedzi[j]:
12         if not odwiedzone[i]:
13             zmienna++
14             skoki[j] = max(skoki[j], dfsik(i, studnia+1))
15
16     if studnia == 0 and zmienna > 1:
17         nozyczki = j
18
19     if skoki[j] <= studnia
20         nozyczki = j
21
22     return skoki[j]
23
24
25 def rozspojnia():
26     dfsik(0, 0)
27     if nozyczki == -1:
28         print("NIE ISTNIEJE WIERZCHOLEK ROZSPOJNIAJACY")
29     else:
30         print(nozyczki)
```

ZAD. 2.

```
1 odwiedzone = [0] * n
2 kolorowanie = [0] * n
3
4 def dfsik(j, kolor):
5     odwiedzone[j] = 1
6     kolorowanie[j] = kolor
7
8     for i in sasiedzi[j]:
9         if not odwiedzone[i]:
10             if not dfsik(i, kolor * -1): return False
11
12         if kolorowanie[i] == kolorowanie[j]:
13             return False
14     return True
```

ZAD. 3.

```
1 odwiedzone = [0] * n
2 kolorowanie = [0] * n
3
4 pom = []
5
6 def dfsik(j):
7     odwiedzone[j] = 1
8     for i in sasiedzi[j]:
9         if not visited[i]: dfs(i)
10
11     pom.append(i)
12
13 def topo():
14     for i in range(len(pom)):
15         kolejnosc.append(pom[n-i])
```

ZAD. 9

```
1 for i in range(n):
2     for j in range(n):
3         for k in range(n):
4             if (d[j][i] < INF && d[i][k] < INF):
5                 nc = d[j][i] + d[i][k]
6                 if d[j][k] > nc:
7                     d[j][k] = nc
8                     p[j][k] = p[j][i] + p[i][k]
```