maths langage

06/02/2019

TP 3

Exercice 1:

```
def parcours(G, a):
    pile = [a]
    atteint = {a}
    while (pile):
        courant = pile.pop()
        for v in G[courant]:
            if not v in atteint:
                atteint.add(v)
                pile.append(v)
    return atteint
```

Exercice 2:

1) parcours(G, 6)

2) 1. chem_existe(G, 6, 1)

   2. Bob: len(parcours(G,6))
      Alice: len(parcours(G,1))

   3. nb_composantes(G)

3) à faire en TD:
   1. chemin(G, 1, ville)
   2. cycle(g, depart)

Exercice 3 :

1) def tresor(G, n):
    return G.node[n]["tresor"]

2) def max_tresor(G, L):
    m = tresor(G, L[0])
    noeud = L[0]
    for i in L:
        if tresor(G, i) > m:
            m = tresor(G, i)
            noeud = i
    return noeud

def parcours_tresor(G, n):
    pile = [n]
    atteint = {n}
    res = []
    while (pile):
        courant = max_tresor(G, pile)
        pile.remove(courant)
        res.append(courant)
        for v in G[courant]:
            if v not in atteint:
                atteint.add(v)
                pile.append(v)
    return res

exercice 4 :

1) def est_courant(G, liste):
    return all([x in [y for y, _ in liste]
                + [y for _, y in liste] for x in G.nodes()])

2) def courant(G):
    return G.edges()

3) def plus_court(G):
    dep = G.nodes()[0]
    pile = [dep]
    atteint = {dep}
    res = set()
    while (pile):
        courant = pile.pop()
        for v in G[courant]:
            if v not in atteint:
                atteint.add(v)
                pile.append(v)
                res.add((courant, v))
    return res