Connexité

vendredi 28 avril 2017 14:56

```
def exploration(graphe, noeud, liste):
    atteignables = [voisin for voisin in voisins(graphe, noeud) if voisin not in liste]
    liste = liste + atteignables
    return len(atteignables) + sum(exploration(graphe, voisin, liste) for voisin in atteignables)
```

Coût en $O(n+m)$ avec n le nombre de nœuds et m le nombre de sommets.

Algorithme comptant les composantes connexes :

```
def composante_connexe(graphe, noeud, liste):
    atteignables = [voisin for voisin in voisins(graphe, noeud) if voisin not in liste]
    liste = liste + atteignables
    return len(atteignables) + sum(exploration(graphe, voisin, liste) for voisin in
atteignables)
def composantes_connexes(graphe):
    vus = set()
    nombre = 0
    for noeud in graphe:
        if noeud not in vus:
            for element in composante_connexe(graphe, noeud, []):
                vus.add(element)
            nombre += 1
    return nombre
```

Coût en $O(n+m)$