

Corrigé du TP Informatique 31

Exercice 1

1. On saisit :

```
>>> a=deque()
>>> a.append(1)
>>> a.appendleft(2)
>>> a.append(3)
>>> a.appendleft(2)
```

2. On vérifie :

```
>>> a.append(a.popleft())
>>> a
deque([2, 3, 4, 1])
>>> a.appendleft(a.pop())
>>> a
deque([1, 2, 3, 4])
```

Exercice 2

1. On saisit :

```
def bfs(S,A,s0):
    F=deque()
    couleur,dist,predec={},{},{ }
    for s in S:
        dist[s]=np.inf
        couleur[s]="blanc"
    F.append(s0)
    dist[s0]=0
    couleur[s0]="gris"
    while len(F)!=0:
        sorg=F.popleft()
        for s in A[sorg]:
            if couleur[s]=="blanc":
                F.append(s)
                dist[s]=dist[sorg]+1
                couleur[s]="gris"
                predec[s]=sorg
        couleur[sorg]="noir"
    return dist,predec,couleur
```

2. On obtient :

```
>>> bfs(S,A,4)
{0: inf, 1: inf, 2: 1, 3: 2, 4: 0, 5: 2, 6: inf, 7: 2, 8: 2},
{8: 2, 2: 4, 3: 2, 5: 2, 7: 2},
{0: 'blanc', 1: 'blanc', 2: 'noir', 3: 'noir',
 4: 'noir', 5: 'noir', 6: 'blanc', 7: 'noir', 8: 'noir'}
```

Exercice 3

1. On saisit :

```
def diametre(S,A):
    diam=0
    s_diam=[]
    for s0 in S:
        dist,predec,couleur=bfs(S,A,s0)
        for cle,val in dist.items():
            if val>=diam:
                if val==diam:
                    s_diam.append([s0,cle])
            else:
                diam=val
                s_diam=[[s0,cle]]
    return diam,s_diam
```

2. On obtient :

```
Graphe orienté
diametre= inf
sommets=
[[1, 0], [1, 6], [2, 0], [2, 1], [2, 6], [3, 0], [3, 1], [3, 6], [4, 0],
 [4, 1], [4, 6], [5, 0], [5, 1], [5, 6], [6, 0], [6, 1], [6, 2], [6, 3],
 [6, 4], [6, 5], [6, 7], [6, 8], [7, 0], [7, 1], [7, 6], [8, 0], [8, 1],
 [8, 2], [8, 3], [8, 4], [8, 5], [8, 6], [8, 7]]

Graphe non orienté
diametre= 3
sommets=
[[1, 5], [1, 7], [1, 8], [3, 6], [4, 6], [5, 1], [5, 6], [6, 3], [6, 4],
 [6, 5], [6, 7], [6, 8], [7, 1], [7, 6], [8, 1], [8, 6]]
```

Exercice 4

1. On saisit :

```
def bipart_dfs(A,s0,color):
    #print(locals())
    F=deque()
    F.append(s0)
```

```
while len(F)!=0:
    sorg=F.popleft()
    for s in A[sorg]:
        if color[s]==color[sorg]:
            return False
        elif color[s]==None:
            F.append(s)
            color[s]=not(color[sorg])
return True
```

2. On saisit :

```
def bipart_init_dfs(S,A):
    color={}
    for s in S:
        color[s]=None
    for s in S:
        if color[s]==None:
            color[s]=True
            if not bipart_dfs(A,s,color):
                return False
    return True
```