

Corrigé du TP Informatique 33

Exercice 1

1. On saisit :

```
def dijkstra(S,A,s0):
    cout,predec,djvu={},{},{ }
    for s in S:
        cout[s]=np.inf
        djvu[s]=False
    cout[s0]=0
    for _ in S:
        smin,cmin=s0,np.inf
        for s in S:
            if not djvu[s] and cout[s]<cmin:
                smin,cmin=s,cout[s]
        djvu[smin]=True
        for s,nu in A[smin]:
            if not djvu[s] and cout[s]>cout[smin]+nu:
                cout[s]=cout[smin]+nu
                predec[s]=smin
    return cout,predec
```

2. On obtient :

```
>>> dijkstra(S,A,0)
{0: 0, 1: 7, 2: 5, 3: 10, 4: 14, 5: 12},
{1: 2, 2: 0, 3: 1, 4: 5, 5: 3}
```

Exercice 2

1. On saisit :

```
def d2(A,B):
    xA,yA=A
    xB,yB=B
    return np.sqrt((xA-xB)**2+(yA-yB)**2)
```

2. On saisit :

```
def Astar_deb_fin(S,A,deb,fin):
    nb=0
    def h(s):
        return d2(s,fin)
    cout,dist,predec,djvu={}, {}, {}, {}
    for s in S:
        cout[s]=np.inf
        dist[s]=np.inf
        djvu[s]=False
    cout[deb]=h(deb)
    dist[deb]=0
    while not djvu[fin]:
        nb+=1
        # recherche elt à cout minimal
        smin,cmin=deb,np.inf
        for s in S:
            if not djvu[s] and cout[s]<cmin: # ici ??
                smin,cmin=s,cout[s]
        djvu[smin]=True
        # relachement chez les successeurs
        for s,nu in A[smin]:
            if not djvu[s] and dist[s]>dist[smin]+nu:
                dist[s]=dist[smin]+nu
                cout[s]=dist[s]+h(s)
                predec[s]=smin
    return cout,predec,nb
```

3. On saisit :

```
def dijkstra_deb_fin(S,A,deb,fin):
    nb=0
    cout,predec,djvu={}, {}, {}
    for s in S:
        cout[s]=np.inf
        djvu[s]=False
    cout[deb]=0
    while not djvu[fin]:
        nb+=1
        smin,cmin=deb,np.inf
        for s in S:
            if not djvu[s] and cout[s]<cmin:
                smin,cmin=s,cout[s]
        djvu[smin]=True
        for s,nu in A[smin]:
            if not djvu[s] and cout[s]>cout[smin]+nu:
                cout[s]=cout[smin]+nu
                predec[s]=smin
    return cout,predec,nb
```

4 Avec un départ en $(2,0)$ par exemple, on obtient

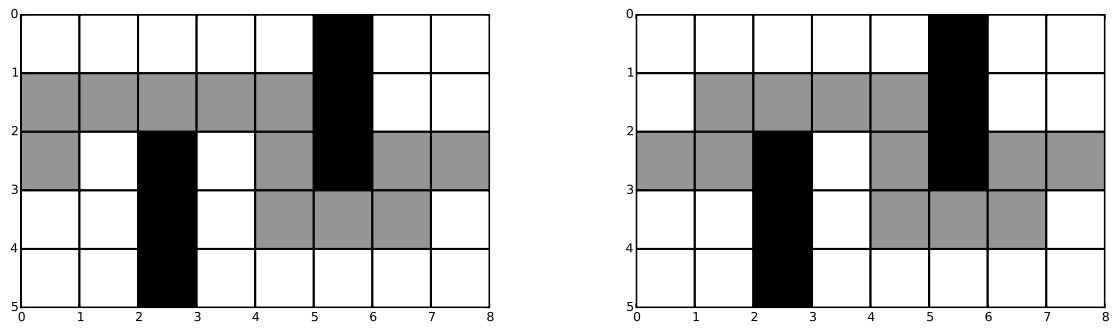


FIGURE 1 – Plus court chemin avec Dijkstra et A*

avec

Dijkstra - nb= 30
A* - nb= 24

Avec un départ en $(0,4)$ par exemple, on obtient

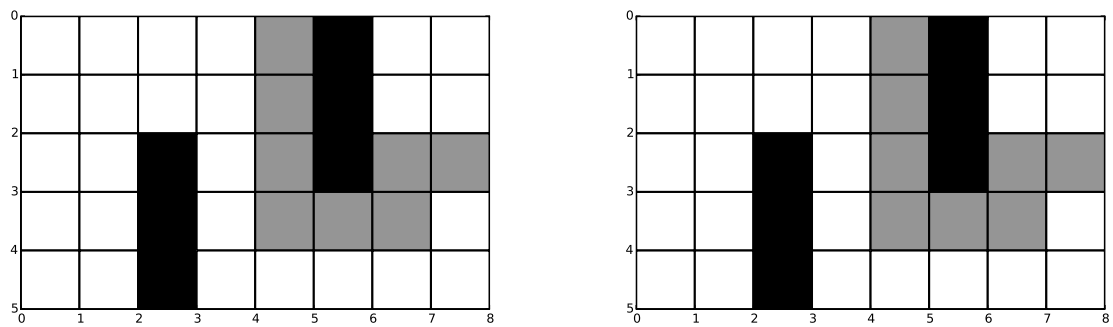


FIGURE 2 – Plus court chemin avec Dijkstra et A*

avec

Dijkstra - nb= 27
A* - nb= 11

Avec ce choix de départ en $(0,4)$, on constate la très nette amélioration de l'algorithme A* vis-à-vis de l'algorithme de Dijkstra.