

## Exercice 2

1.  $v = 6$
2.

```
def plateau_init(n, m, cartes):
    shuffle(cartes)
    plateau = []
    for i in range(n):
        plateau.append([cartes[i+j*n] for j in range(m)])
    return plateau
```
3.

```
def cartes_voisines(n, m, i, j):
    voisines = []
    for i2 in range(i-1, i+2):
        for j2 in (j-1, j+2):
            if (i2, j2) != (i, j) and i2 in range(n) and j2 in range(m):
                voisines.append((i2,j2))
    return voisines
```
4.

```
e1 = 9 + 8 + 6 + 7 = 30
e2 pas une chaine
e3 pas une chaine
```
5.

```
def chaine_value(plateau, chaine):
    s = 0
    for i,j in chaine:
        s += plateau[i][j]
    return s
```
6.

En faisant un parcours en largeur, la chaine qui aura la valeur recherchée sera forcément la plus courte.
7.

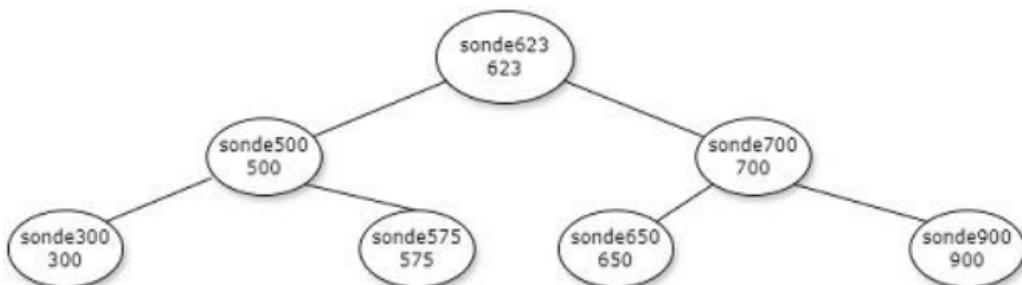
```
def explore(plateau, cible):
    n, m = len(plateau), len(plateau[0])
    a_visiter = file_init()
    for i in range(n):
        for j in range(m):
            file ajoute(a_visiter, [(i,j)])
    while not file_est_vide(a_visiter):
        chemin = file_retreive(a_visiter)
        if chaine_value(plateau, chemin) == cible:
            return chemin
        dernier = chemin[-1]
        i0, j0 = dernier
        for i, j in cartes_voisines(n, m, i0, j0):
            if (i, j) not in chemin:
                file ajoute(a_visiter, chemin + [ (i,j) ])
    return None
```

8. On crée une nouvelle liste `tous_chemins` et à la ligne 11, au lieu de faire un `return chemin`, on fait un `tous_chemins.append(chemin)`

### Exercice 3

```

1. enregistrement['latitude']
2. ('2024-06-27','23:36:01')
3. 4 pour len(frames) et 5 pour len(frames[1])
4.
    def detecter_anomalie(d):
        return d['altitude'] < 0 or d['altitude'] > 35000
5.
    def liste_num_serie(frames):
        n_serie = []
        for e in frames:
            if e['num_serie'] not in n_serie:
                n_serie.append(e['num_serie'])
6.
    def distance_totale(dep):
        total = 0
        for i in range(1,len(dep)):
            total += distance_haversine(dep[i-1], dep[i])
        return total
7.
sonde623 = Sonde(623,38.38825, 27.09004, '2024-06-27', None, None)
8.
```



9.  
ordre infixe