



**GROUPE 1**



**EBEC 2021 -**

**LOCALISER DES ARBRES EN  
LANGUAGE NATUREL**

Sujet de AIOS SH



## *Introduction*

Présentation de nos outils et de l'équipe

## *Modèle de l'algorithme*

Présentation du fonctionnement de l'algorithme pour chaque objectif.

## *Workflow*

Détail des fonctions impliquées dans le processus.

## *Cas limites*

Présentation des cas problématiques

## *Conclusion*



## Introduction

Algo codé en **Python**

Fait appel à **OpenStreetMap**

**Git** pour travailler en équipe

```
def coord_to_inter(liste_coord, lat, lon):
    dist=[]
    inter_street=[]
    inter_street_coord=[]
    for inter in liste_coord:
        #Calcul des distances entre l
        d=(inter[0]-lon)**2 + (inter
        dist.append(d)
    while len(inter_street)<1 :
        #On regarde l'intersection la
        print('nouveau candidat')
        try:
            candidat=liste_coord[dist
        except ValueError:
            inter_street.append(main_
            inter_street_coord.append
            inter_street.append(main_
            inter_street_coord.append
            print('aucune intersection
```

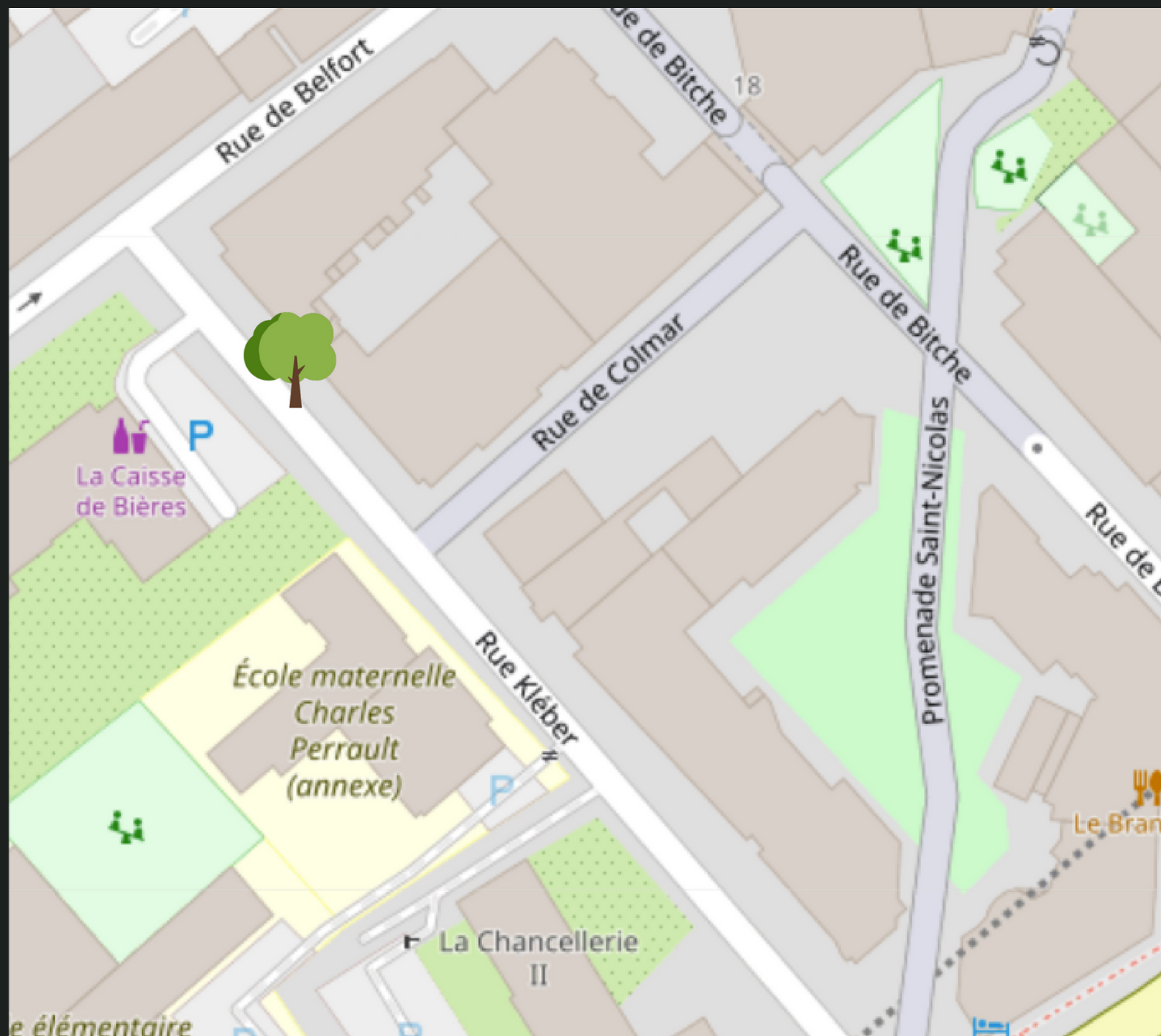




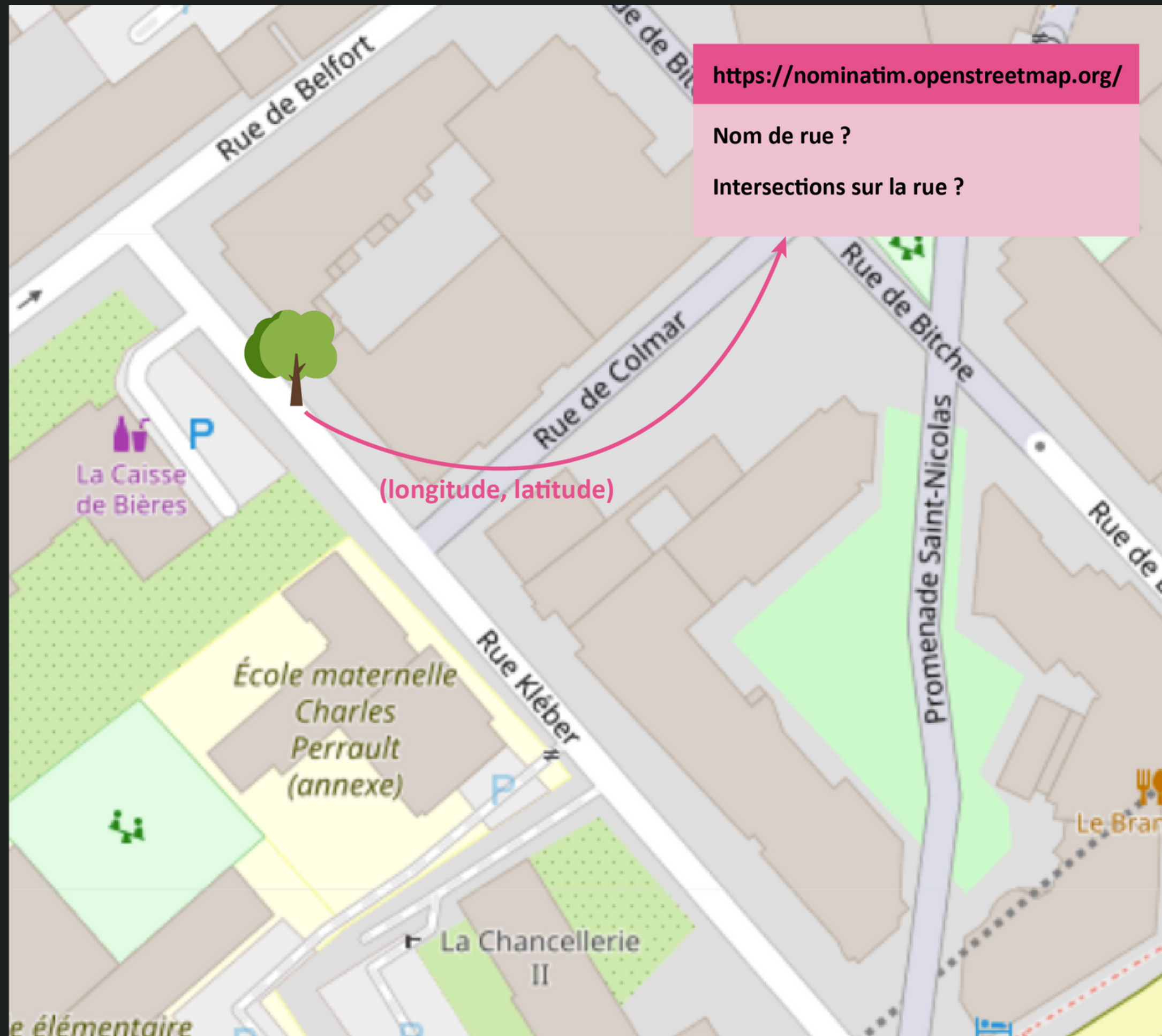
# MODÈLE DE L'ALGORITHME

*objectif 1*



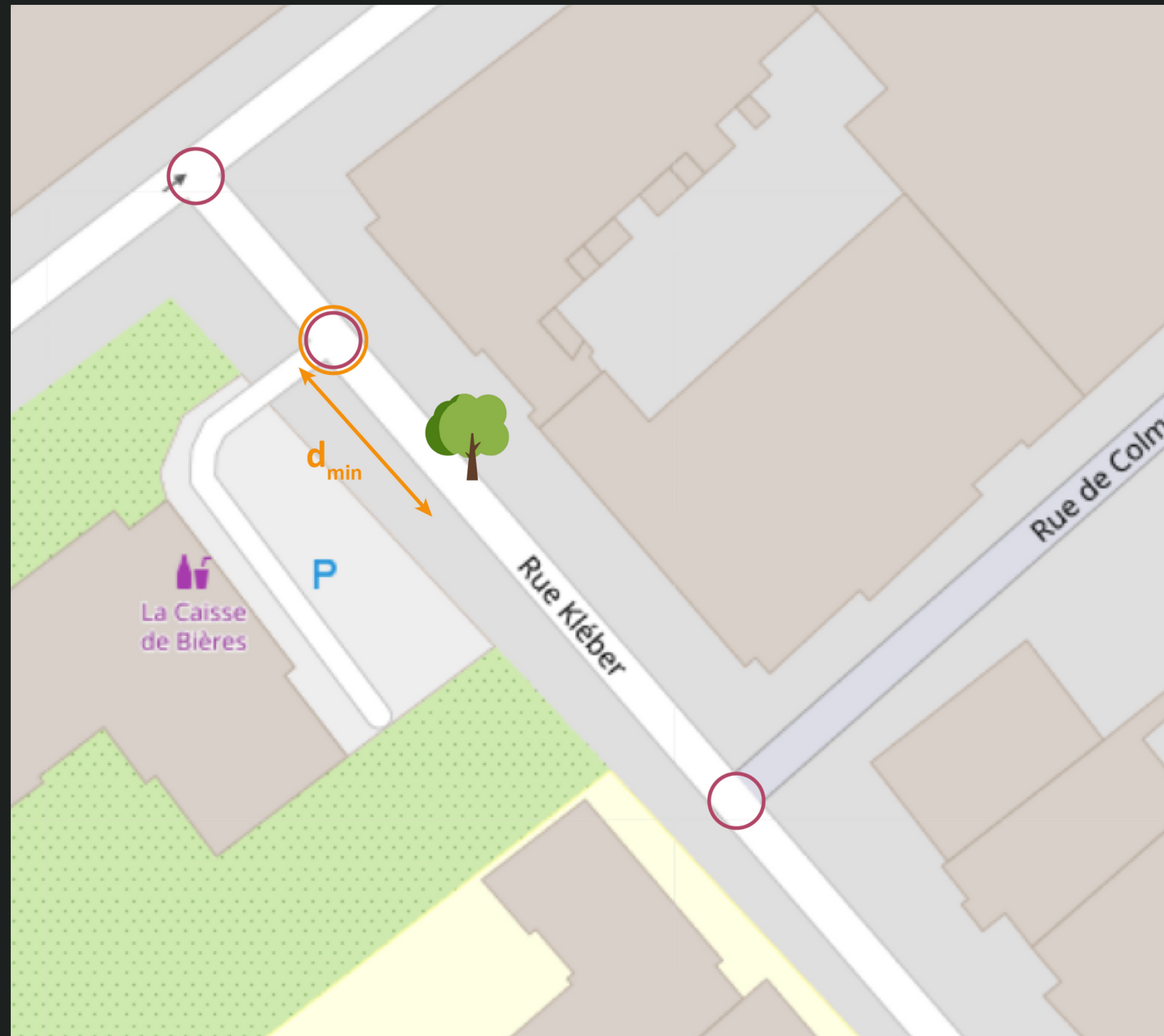








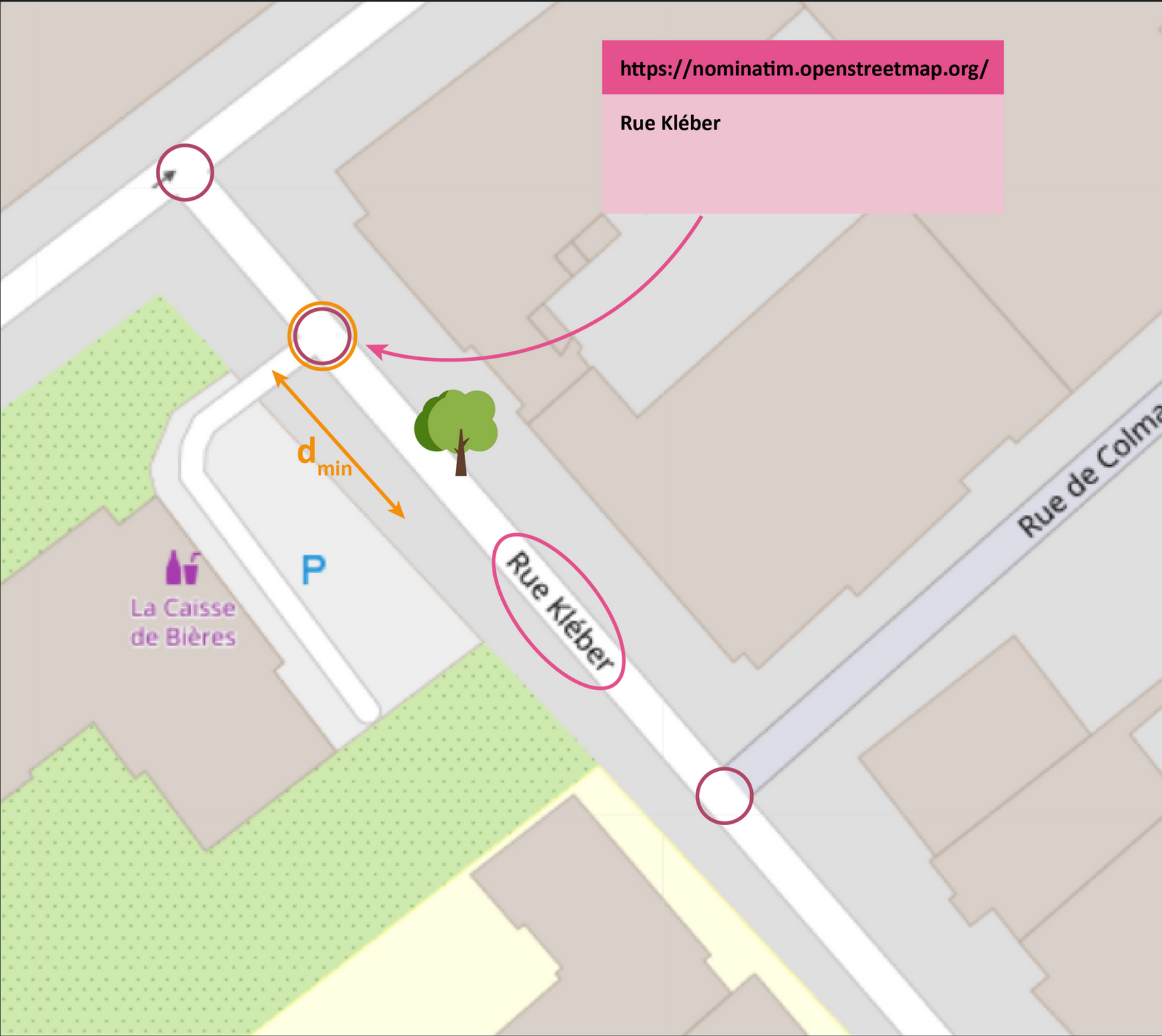








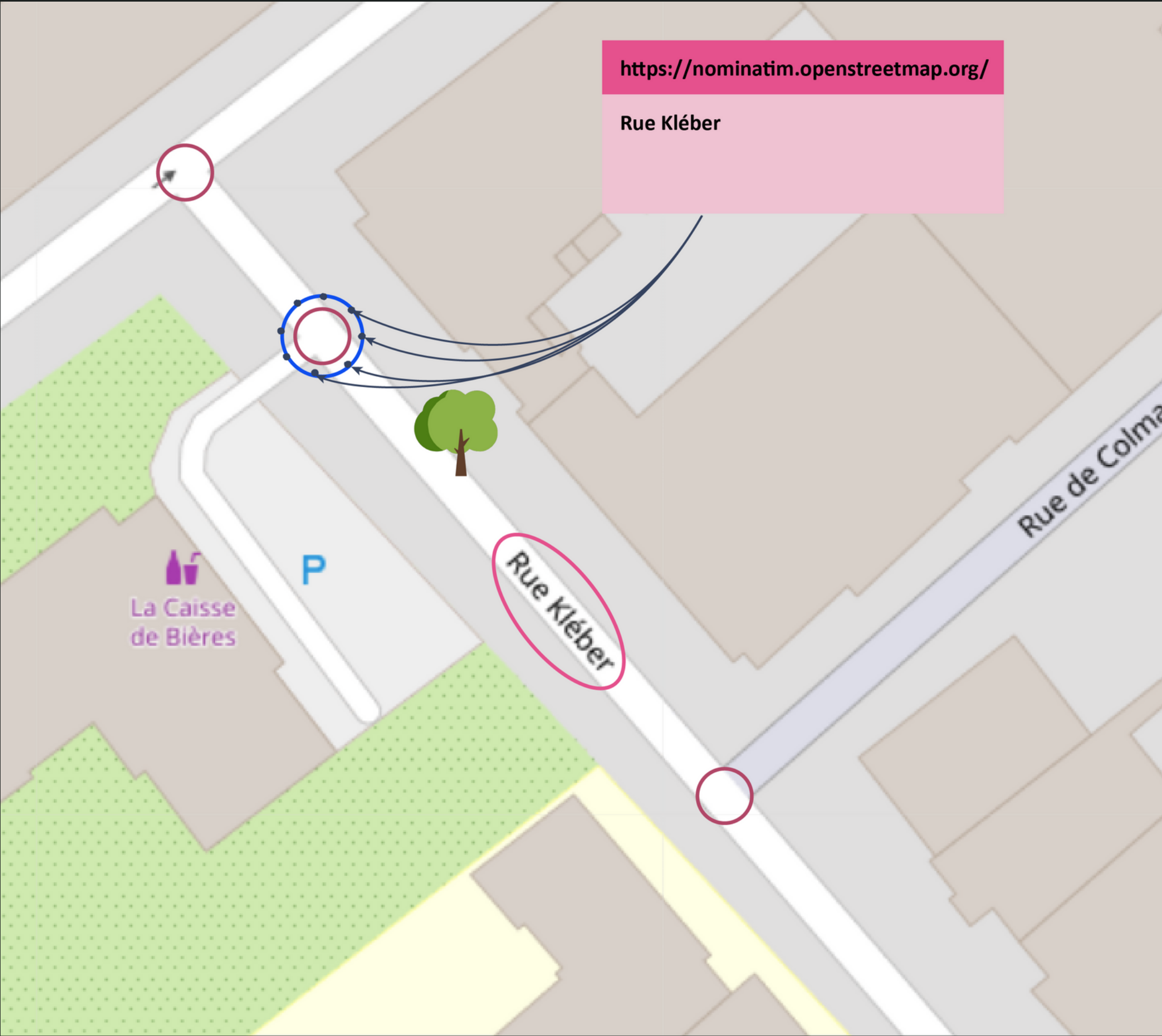
<https://nominatim.openstreetmap.org/>  
Rue Kléber







<https://nominatim.openstreetmap.org/>  
Rue Kléber



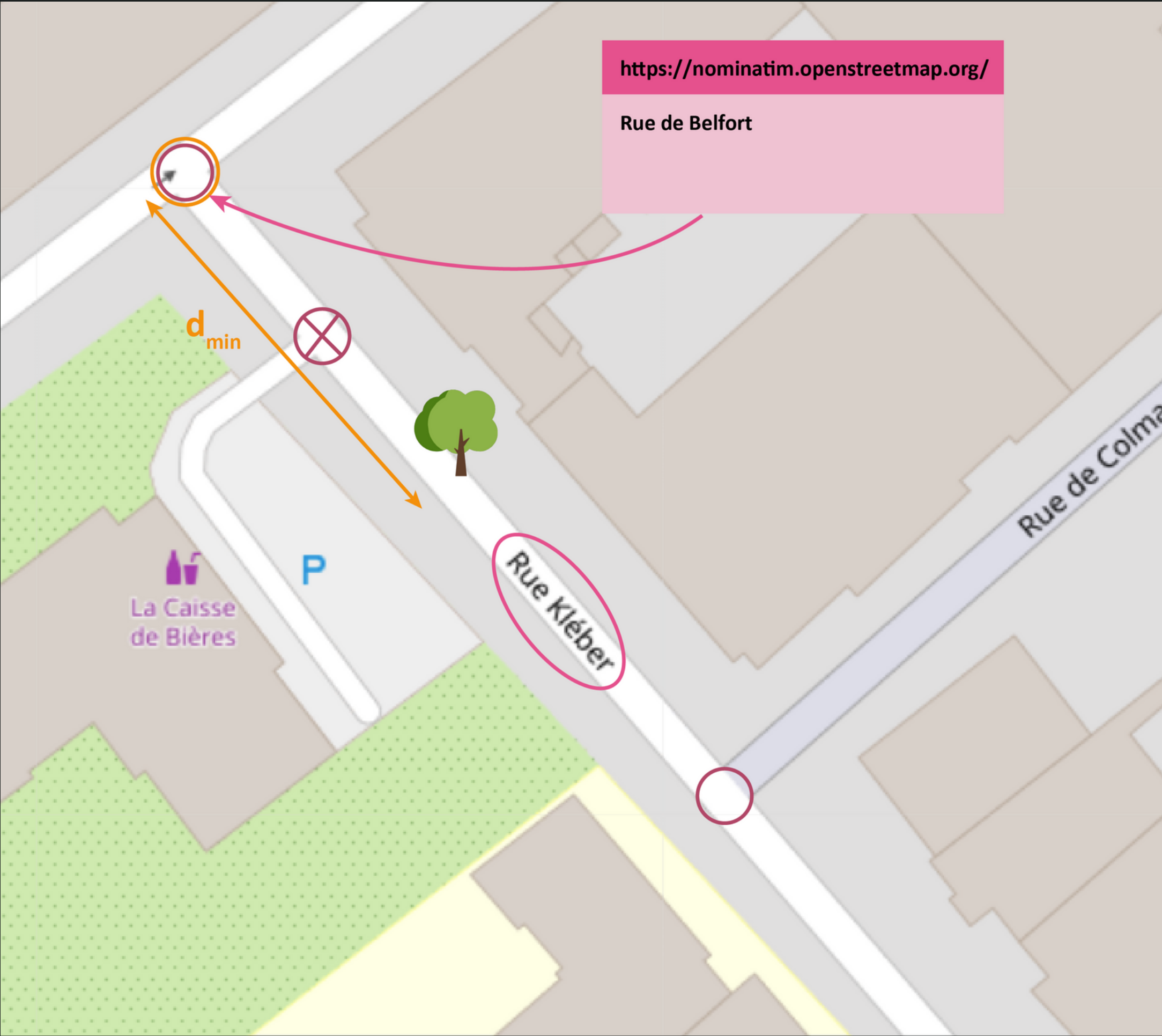


<https://nominatim.openstreetmap.org/>

Rue Kléber



<https://nominatim.openstreetmap.org/>  
Rue de Belfort

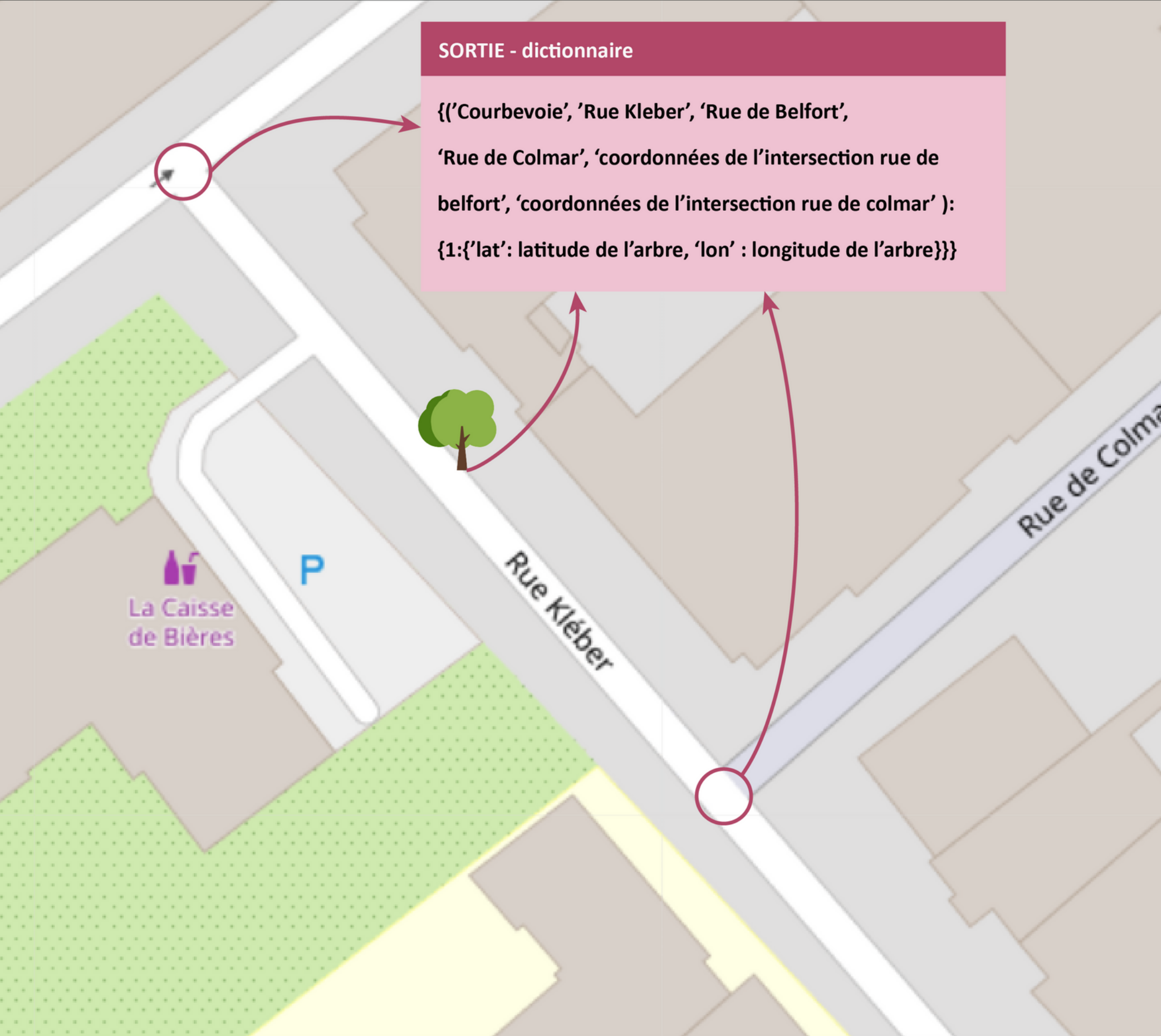












**SORTIE - dictionnaire**

```
{('Courbevoie', 'Rue Kleber', 'Rue de Belfort',  
  'Rue de Colmar', 'coordonnées de l'intersection rue de  
  belfort', 'coordonnées de l'intersection rue de colmar' ):  
  {1:{'lat': latitude de l'arbre, 'lon' : longitude de l'arbre}}}
```



# MODÈLE DE L'ALGORITHME

*objectif 2*



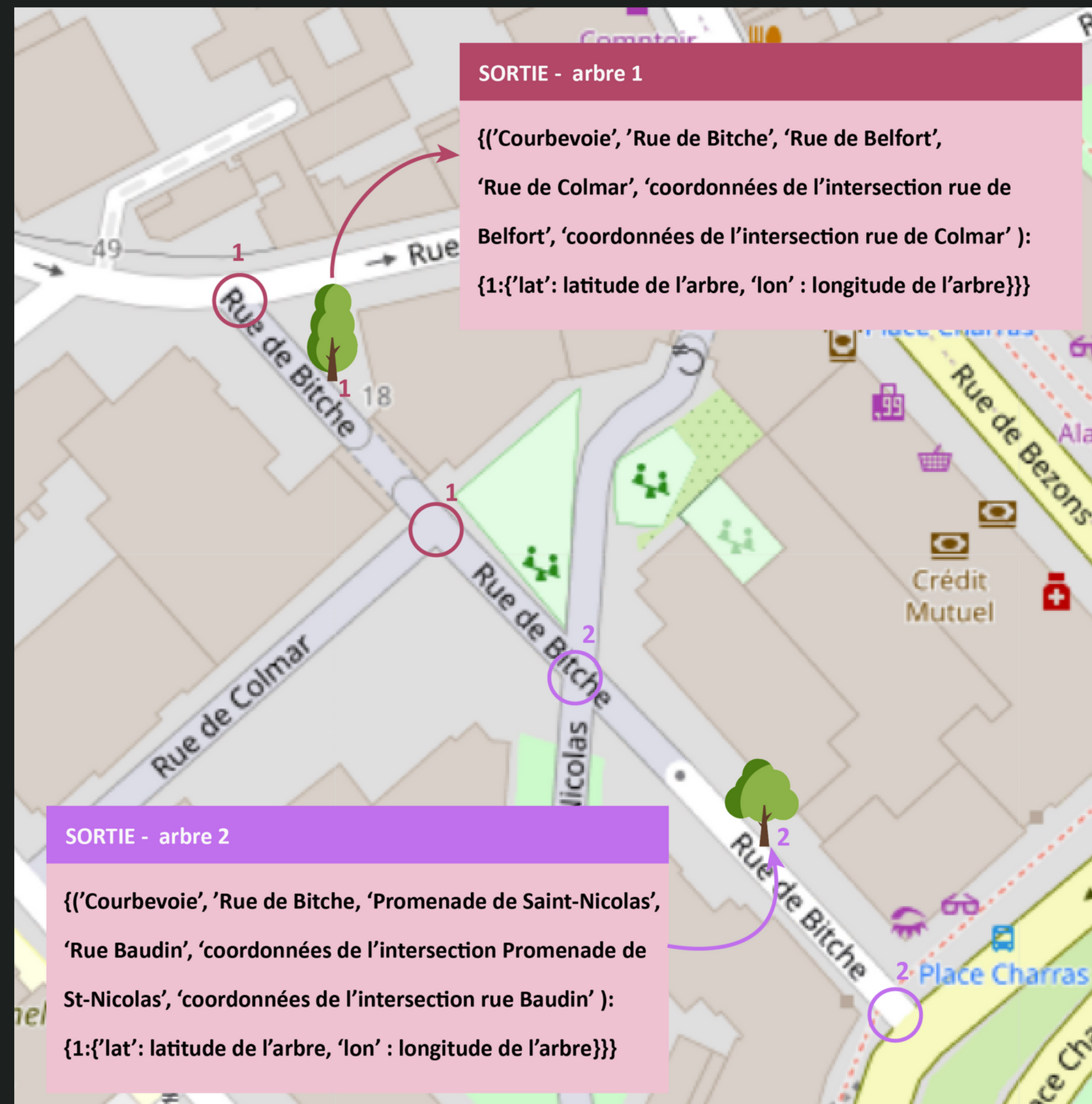






# MODÈLE DE L'ALGORITHME

*objectif 3*



**SORTIE - arbre 1**

{{('Courbevoie', 'Rue de Bitché', 'Rue de Belfort',  
'Rue de Colmar', 'coordonnées de l'intersection rue de  
Belfort', 'coordonnées de l'intersection rue de Colmar' ):  
{1:{'lat': latitude de l'arbre, 'lon' : longitude de l'arbre}}}}

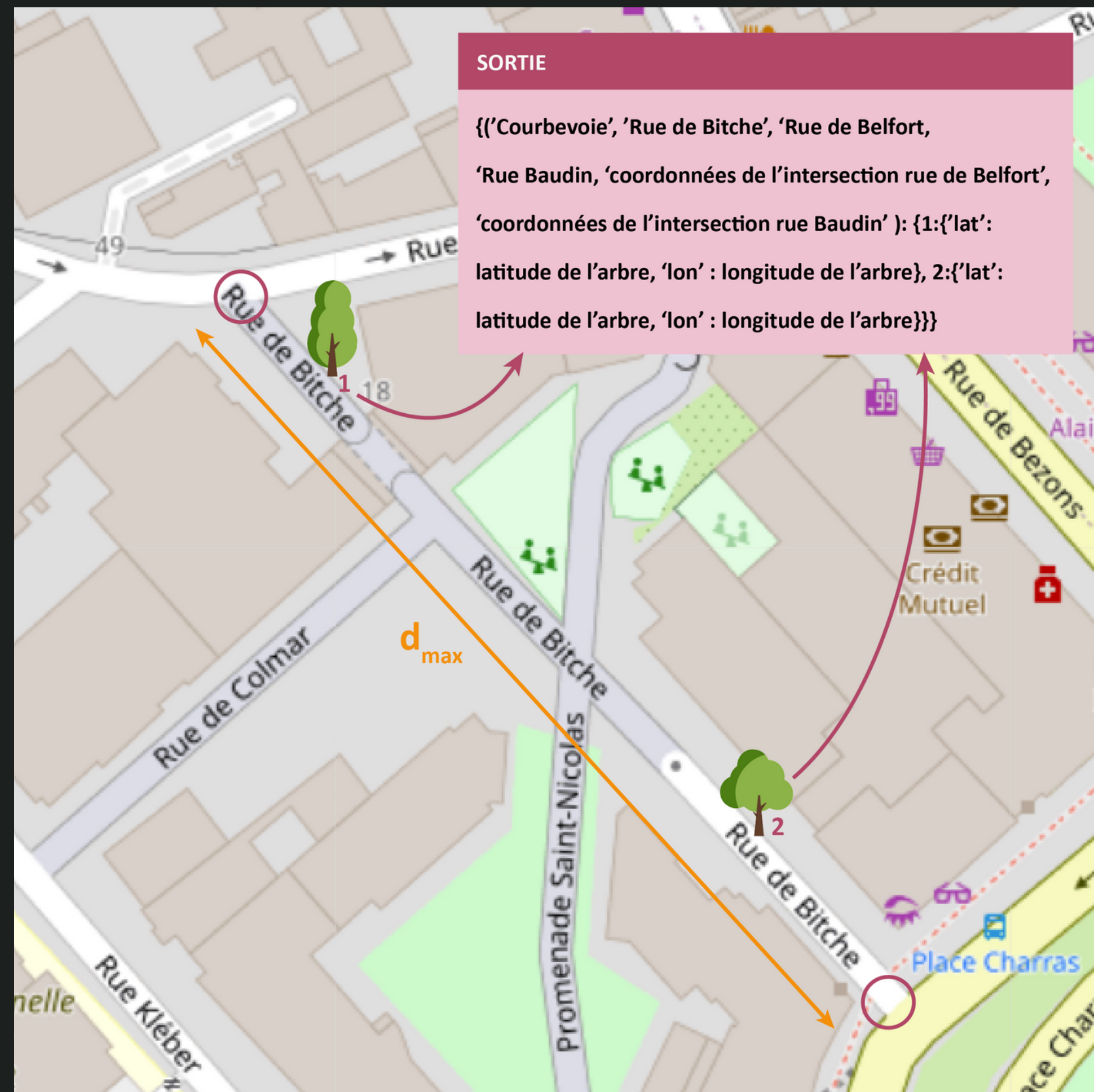
**SORTIE - arbre 2**

{{('Courbevoie', 'Rue de Bitché', 'Promenade de Saint-Nicolas',  
'Rue Baudin', 'coordonnées de l'intersection Promenade de  
St-Nicolas', 'coordonnées de l'intersection rue Baudin' ):  
{1:{'lat': latitude de l'arbre, 'lon' : longitude de l'arbre}}}}



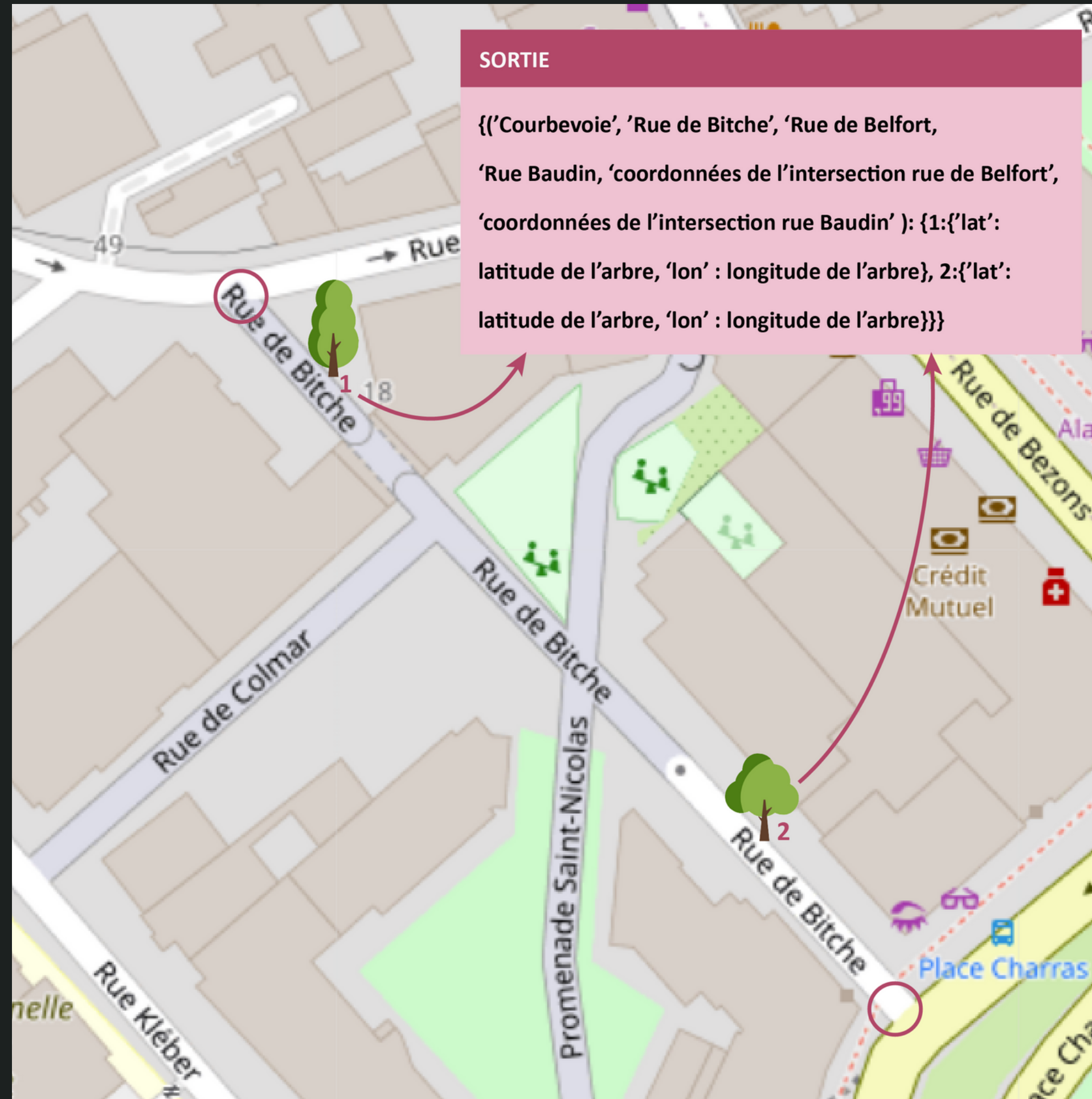






## SORTIE

{{('Courbevoie', 'Rue de Bitche', 'Rue de Belfort',  
'Rue Baudin', 'coordonnées de l'intersection rue de Belfort',  
'coordonnées de l'intersection rue Baudin' ): {1:{'lat':  
latitude de l'arbre, 'lon' : longitude de l'arbre}, 2:{'lat':  
latitude de l'arbre, 'lon' : longitude de l'arbre}}}}



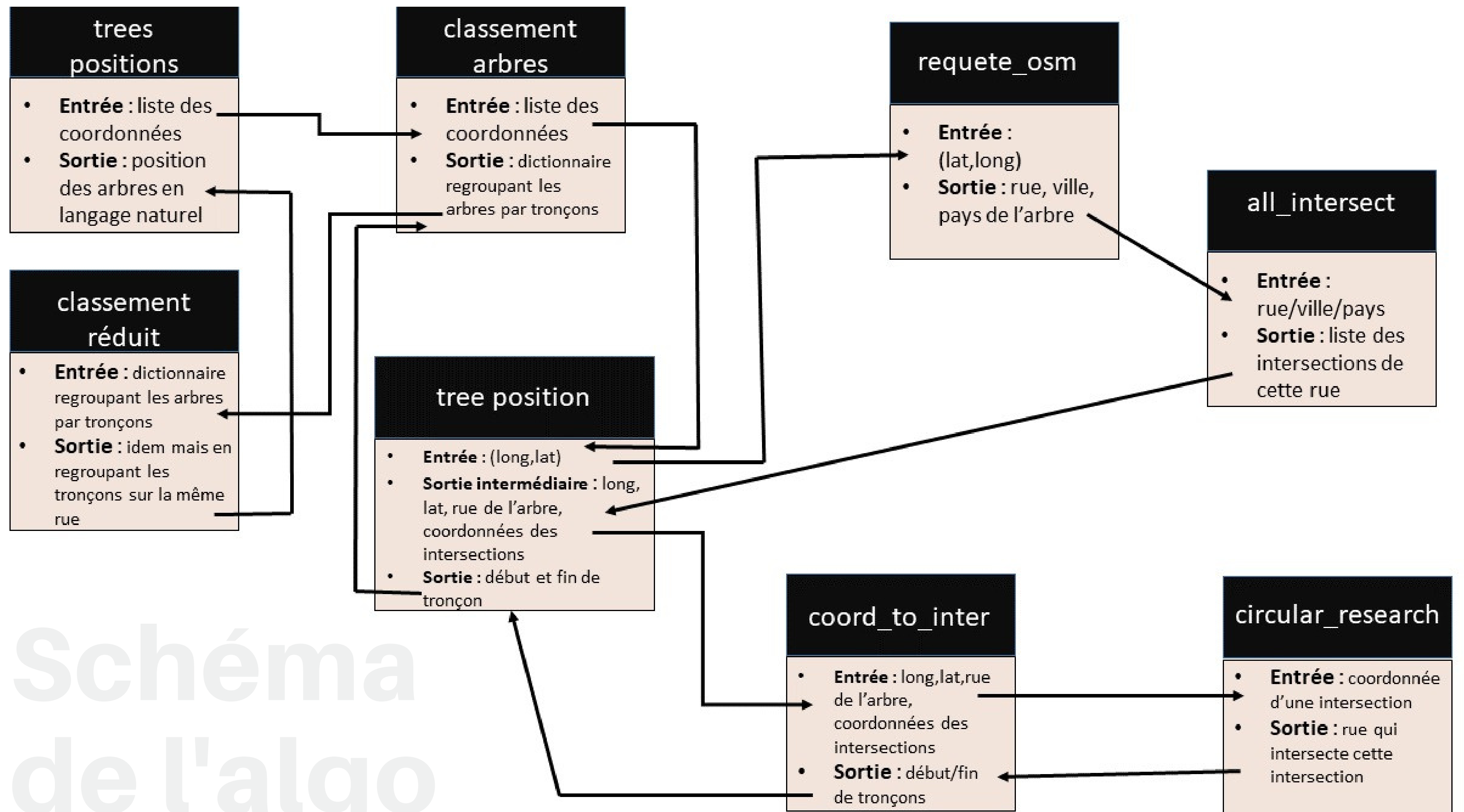


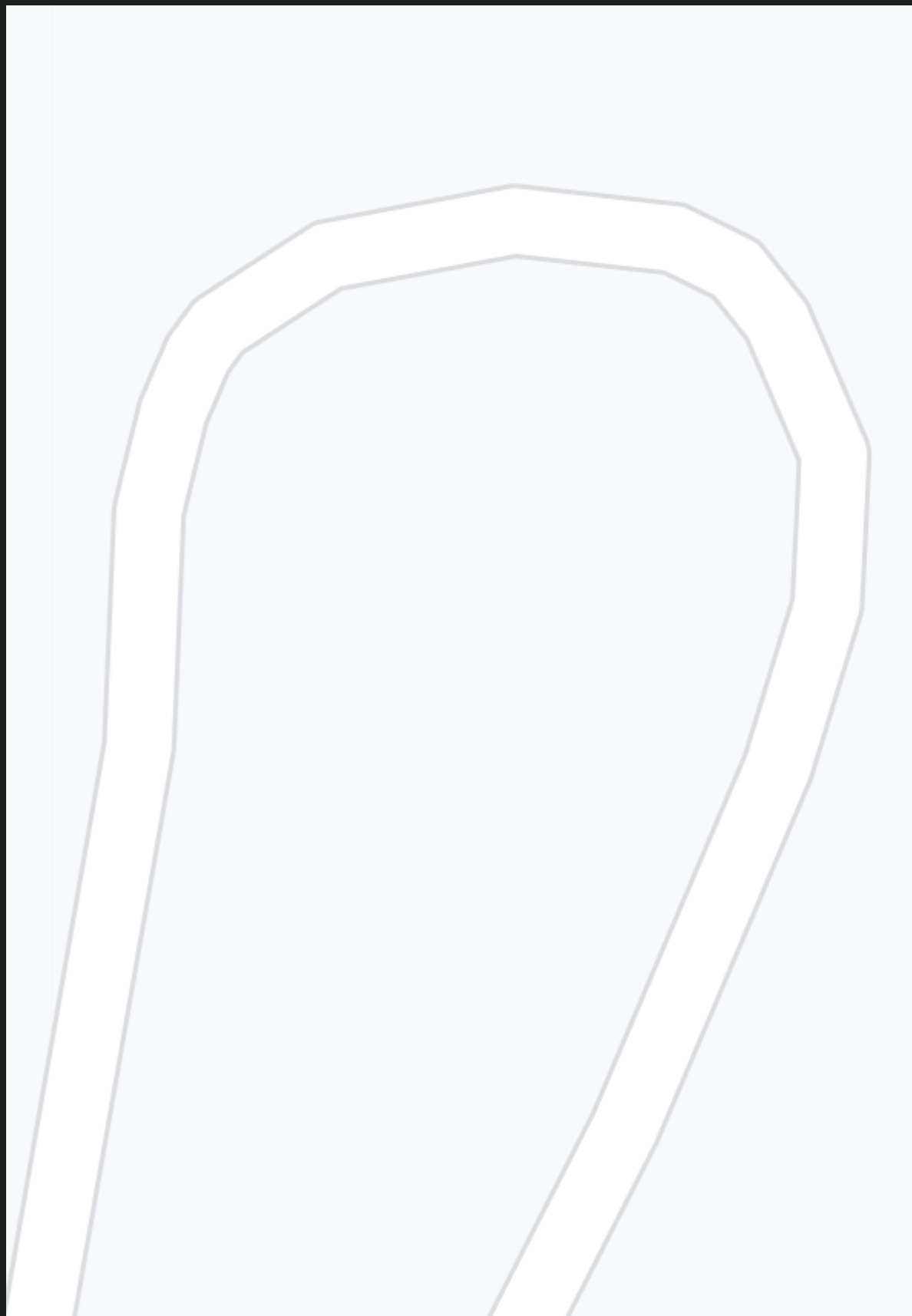


# MODÈLE DE L'ALGORITHME

*Workflow*







# LIMITES DE L'ALGORITHME

Route qui fait un **demi-tour**

*L'**angle** entre les intersections et l'arbre sera **aigu**, il faudrait donc traiter différemment ce cas particulier.*





*Impasses*



*Avenue découpées en  
deux*



*Toutes les routes sans  
virage limite*

# CONCLUSION



4

*Objectifs remplis*

1

*Cas limite*

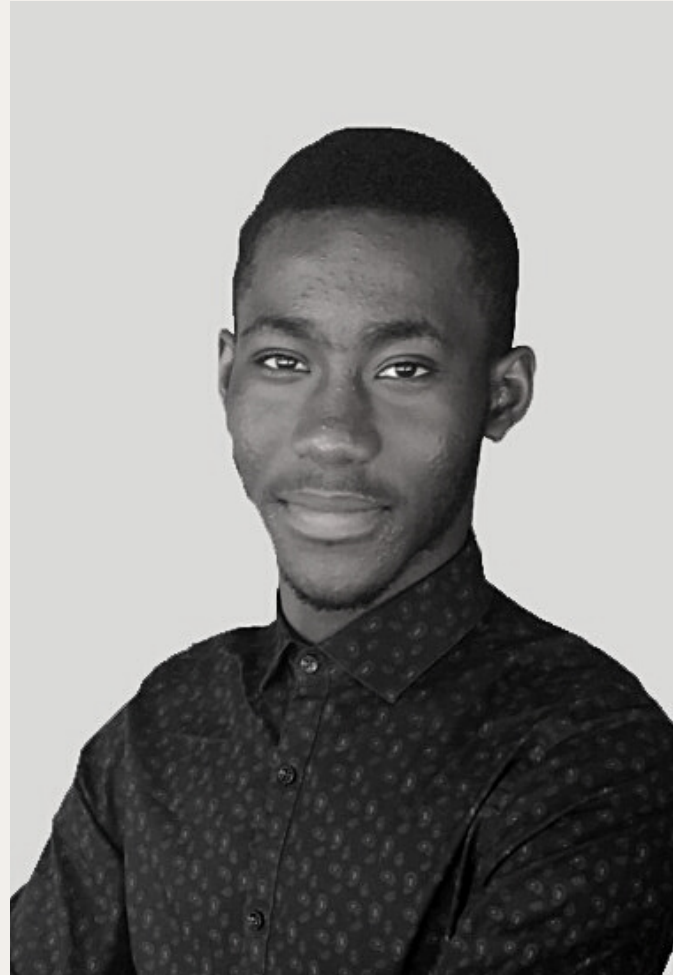
CONCLUSION

# L'équipe



*Léna Aix*

Etudiante en  
deuxième année à  
Centrale Supélec



*Mamadou Diko*

Etudiant en  
deuxième année à  
Central



*Servane  
Gouillou*

Etudiante en  
deuxième année à  
Centrale Supélec



*Alexandre  
Pradeilles*

Etudiant en  
deuxième année à  
Central



CentraleSupélec