Commentaires.

Marquervoisin() était surtout un prétexte à créer une matrice d'adjacence et à commencer à réfléchir au parcours systématique des sommets d'un graphe.
A sa complexité aussi.

En s'inspirant de https://nicolasj.developpez.com/articles/file/
https://nicolasj.developpez.com/articles/pile/
par exemple.

Le parcours en largeur est une première extension propre à ce parcours en utilisant une file d'attente de type FIFO.

Pour cela, nous maintenons dans une file d'attente f l'ensemble des sommets marqués dont nous n'avons pas encore traité les voisins. Une étape de la recherche consiste à sélectionner l'un de ces sommets en le retirant en tête de f, à marquer tous ses voisins (non encore marqués) et à les ajouter en queue de f.

Le tableau **pred** permet de garder le prédécesseur ceci afin de tracer plus tard le plus court chemin à l'envers de n'importe quel sommet au sommet source s.

Avec la recherche en largeur, nous avons toutes les cartes en main pour écrire un algorithme calculant la longueur de tous les plus courts chemins depuis un sommet s. Il suffit au cours de la visite de mettre à jour la longueur **l** pour chaque sommet. La longueur **l[y]** du sommet y est calculé comme la longueur **l[x]** de son voisin x depuis lequel il est visité, plus 1. A la fin de l'algorithme, les longueurs sont les distances minimales de tous les sommets depuis le sommet s.