Theorie des graphes

À quoi sert cette l'application ?

Cette application est faite pour rendre votre vie plus facile lorsque vous traitez des graph et des algorithmes sur des graphs, il couvre une variété d'algorithmes tels que BFS DFS Dijkstra Prime Kosaraju bellman-ford, et il vous donne le résultat d'un algorithme sur un graph sous forme d'un grpah

Code Source:

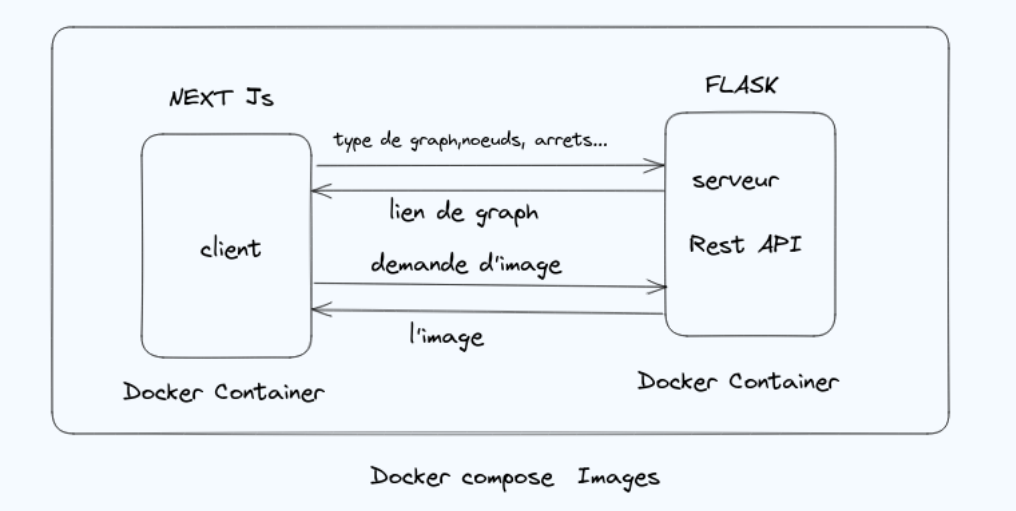
GitHub - sohaibMan/GraphTheory: Graph theory is the study of graphs that concerns with the relationship among edges and vertice, and in this proj

Graph theory is the study of graphs that concerns with the relationship among edges and vertice, and in this project I have implement a various number of algorithmes s

https://github.com/sohaibMan/GraphTheory.git

Comment ça marche ?

Cette application est une application web réalisée avec l'architecture REST (client ,serveur) avec Flask (Python) + Networkx (pour la gestion des graphs)+ Matplotlib(Dessin des graphs) en back end et NextJs + React +TypeScript + Regex (valider l'entrée utilisateur) + React Query (gestion de l'état de l'application) pour le front end et le docker pour le déploiement



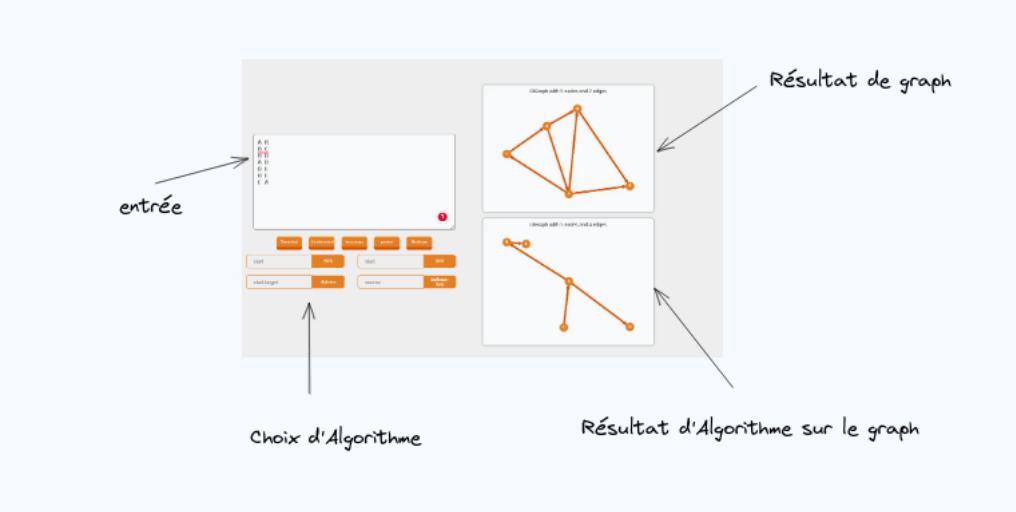
Comment l'installer ?

```
git clone https://github.com/sohaibMan/GraphTheory.git

cd GraphTheory


###################################
## for docker users
###################################
docker compose up
```

```
## to stop it
docker compose down
## to build it
docker compose up --build

####################################
## to runt locally without docker(if you don't have docker installed)
####################################
## install the dependencies (front end)
cd nextjs && npm i
## install the dependencies (backend end)
cd ../flask && pip3 install -r requirements.txt
## note if u run on linux you need to give the app permission to write and read and delete to tmp_output folder if needed
chmod 770 tmp_output
## run the back end
python3 server.py
## run the front end
cd ../next && npm run dev
## now open http://localhost:3000 in your browser
```

Comment utiliser l'application ?

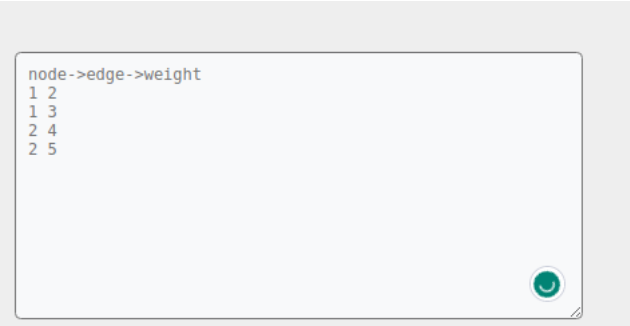
Cette application est conviviale et très simple à utiliser , il suffit de suivre ces étapes pour l'utiliser



1- **Dessiner pour voir votre graphique**

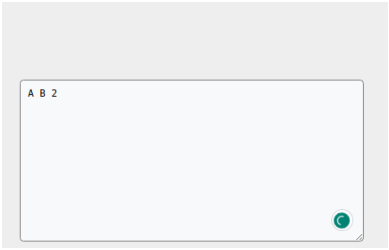
vous avez cette zone de texte pour taper vos bords et nœuds de graphique dans ce format

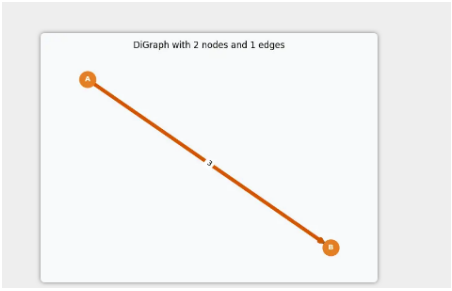
noeud arête le poids (poids est facultatif)



Example :

- **A B 2** sera interprété comme on a 2 noeuds A et B et un arrete A→ B de poids 2 (avec ordre)



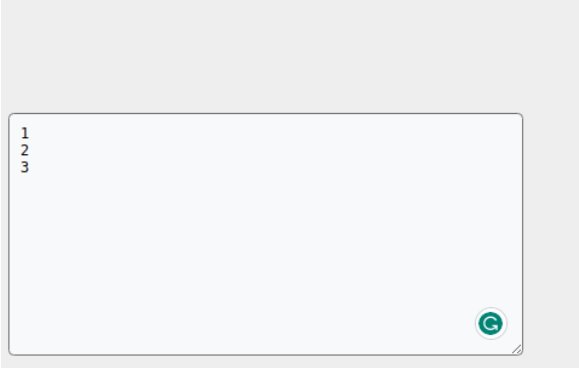


- 1

  2

  3

  sera interprété comma on a 3 noeuds 1 2 3

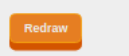


2- choisir un algorithme pour l'appliquer et saisir les valeurs requises

Astuces:

- Pour Changer l'orientation du graph clicker sur l'un des buttons:

- si le graph n'est pas bien dessiné, vous pouvez le redessiner à l'aide de ce bouton

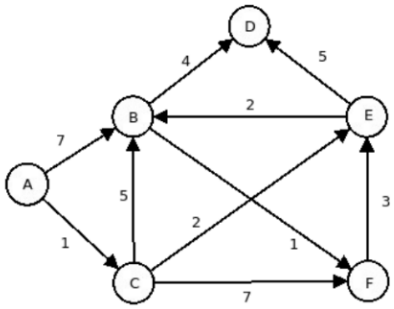


Example de l'algorithme de Dijkstra :

Definition: L'algorithme de Dijkstra est un algorithme pour trouver les chemins les plus courts entre les nœuds dans un graphique pondéré

Exercice

Donner la trace de l'algorithme de Dijkstra pour déterminer l'arborescence des plus courts chemins depuis le sommet A dans le graphe G orienté pondéré ci-dessous :



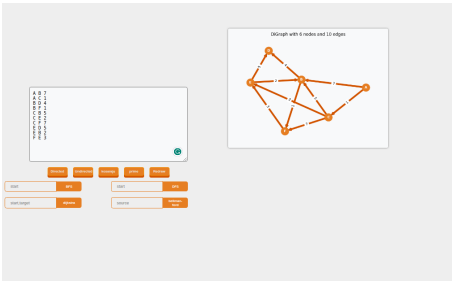
Les Etapes :

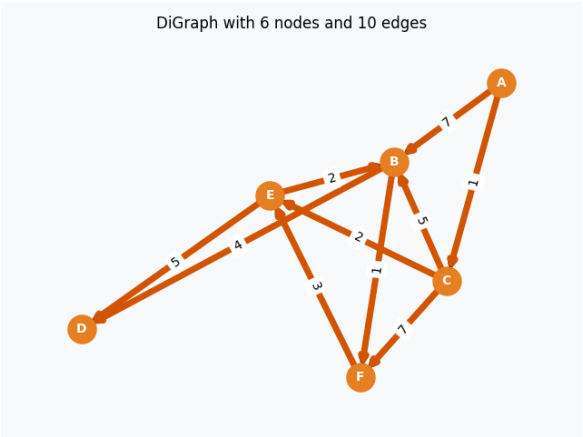
1. **Représentation graphique sous forme d'une liste**

Notes : Nous supposons que le nœud A est 1 B est 2 et vice versa

```
A B 7
A C 1
B D 4
B F 1
C B 5
C E 2
C F 7
E D 5
E B 2
F E 3
```

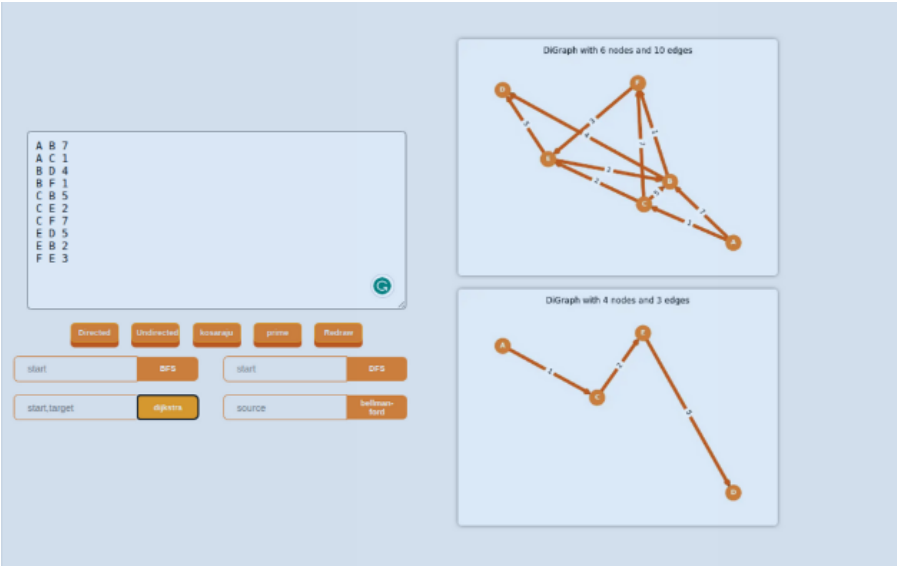
L'interface





2. le choix d'un nœud de départ et noe d'extrémité

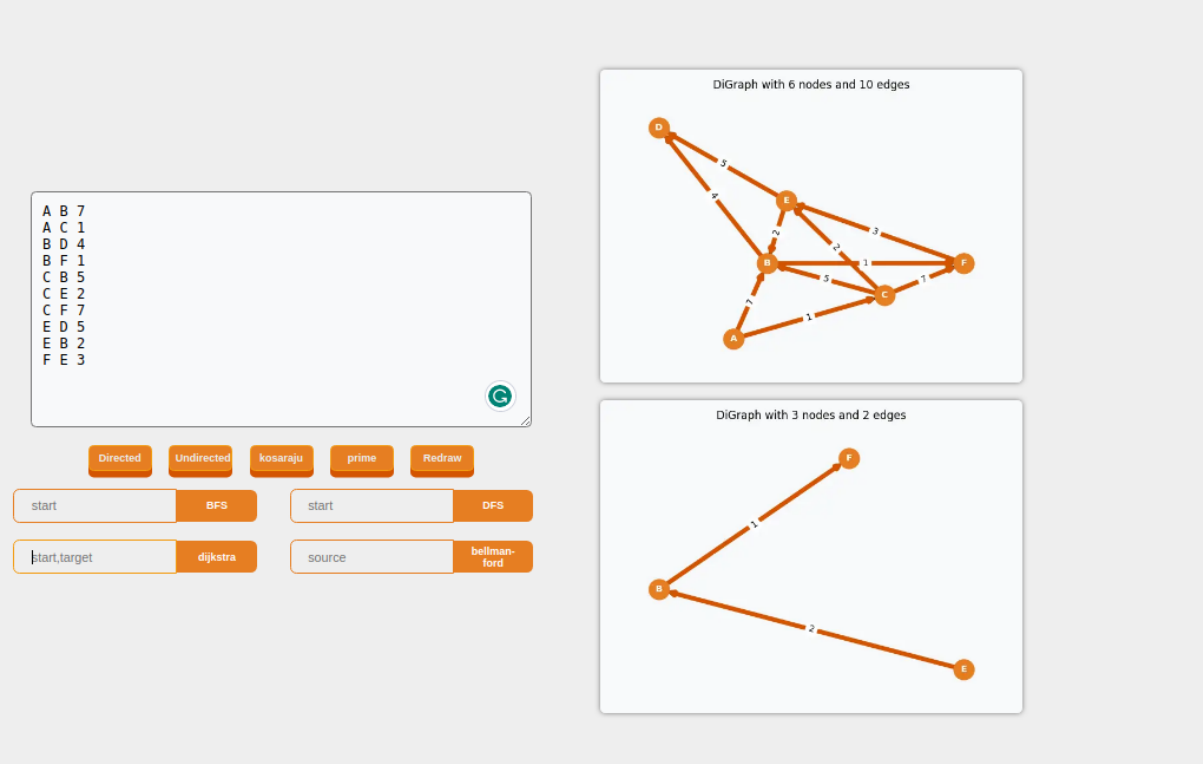
Exemple: A,D



Explication:

- le chemin le plus court entre A et D est A→C→E→D avec le cout total 8

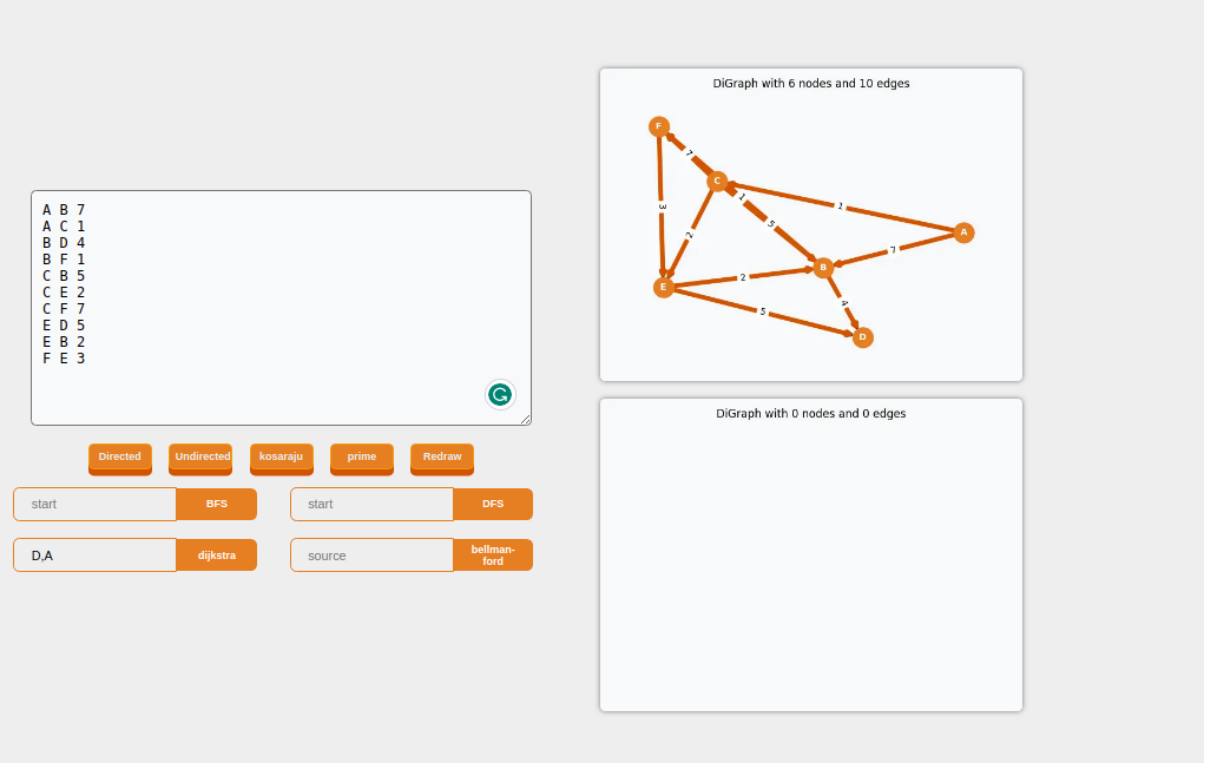
Exemple: E,F



Explication:

- le chemin le plus court entre E et F est E→B→F avec le cout total 3

Exemple : D,A



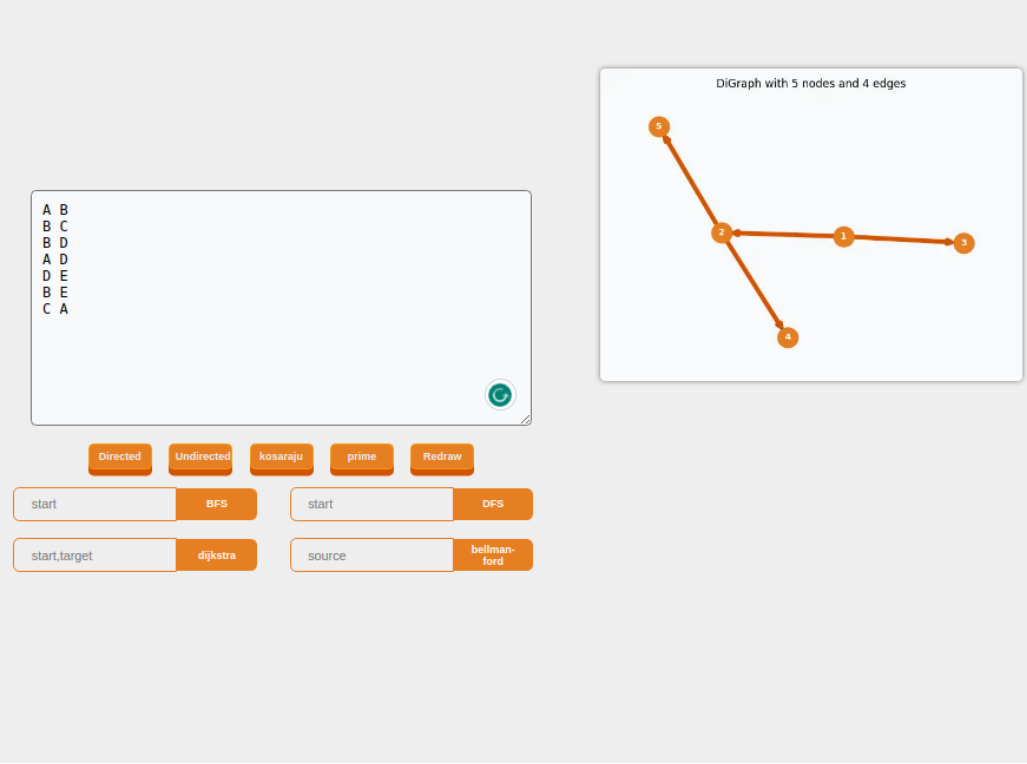
Explication:

- il ya pas de chmen entre D et A

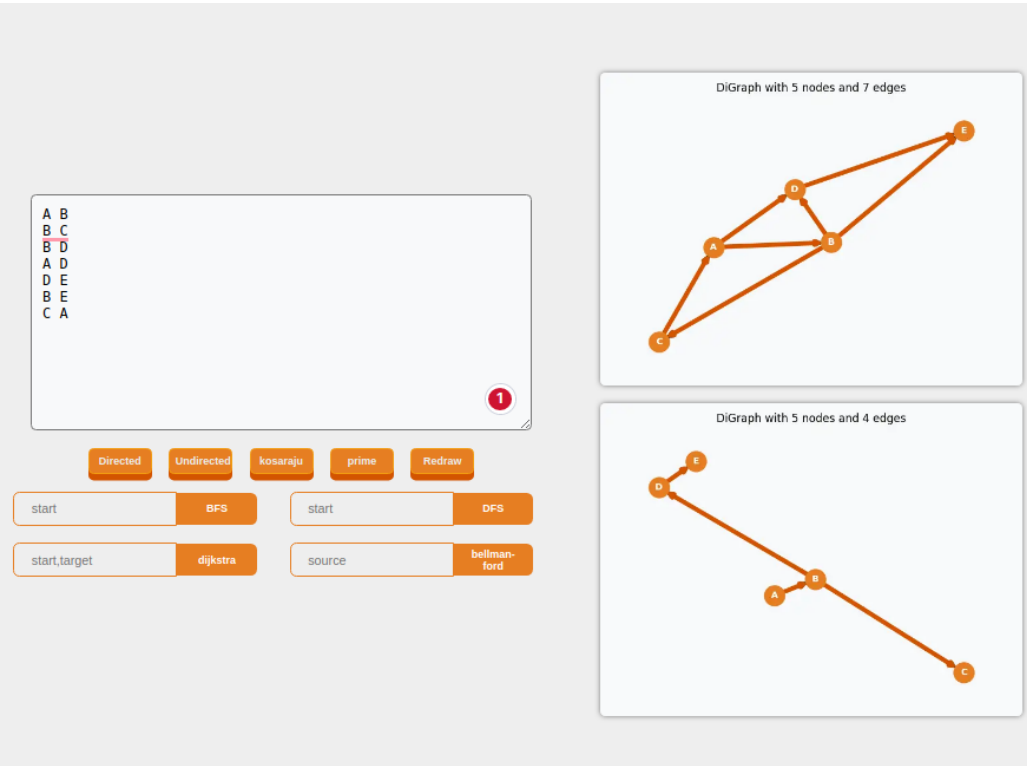
Example de l’algorithme de BFS / DFS:

pronouns par exemple le graph suivante: (graph simple orienté pondéré)

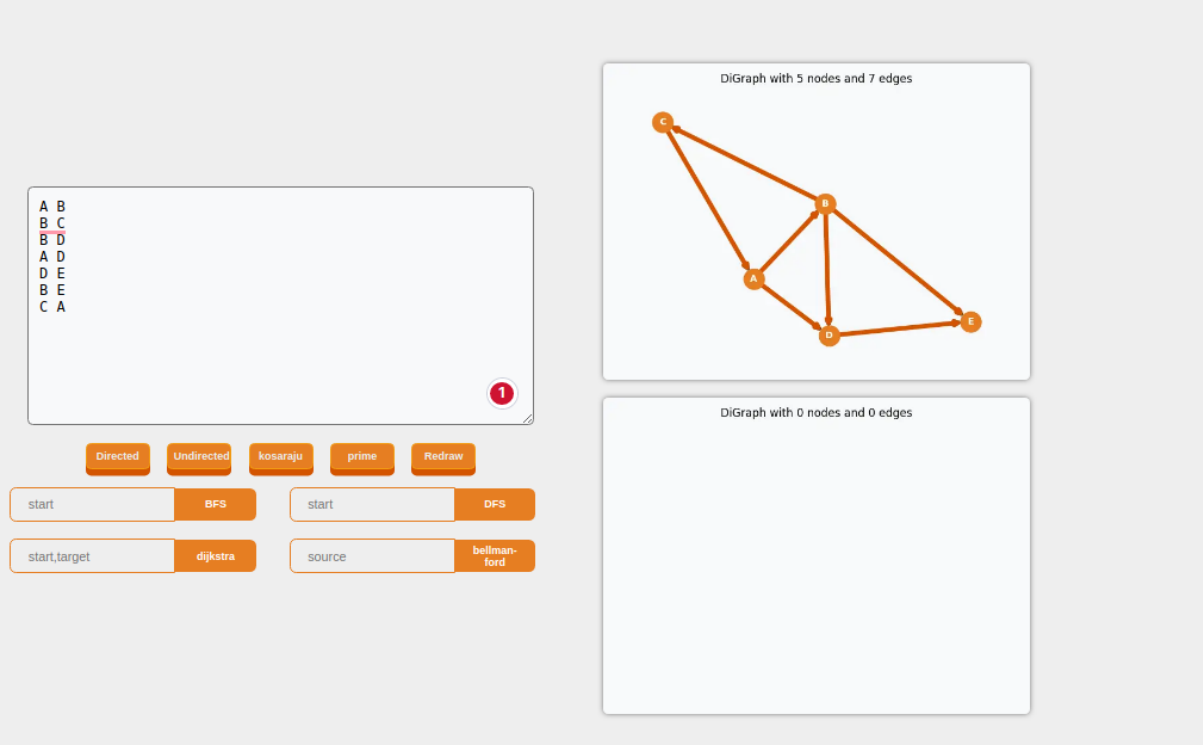
```
A B
B C
B D
A D
D E
B E
C A
```



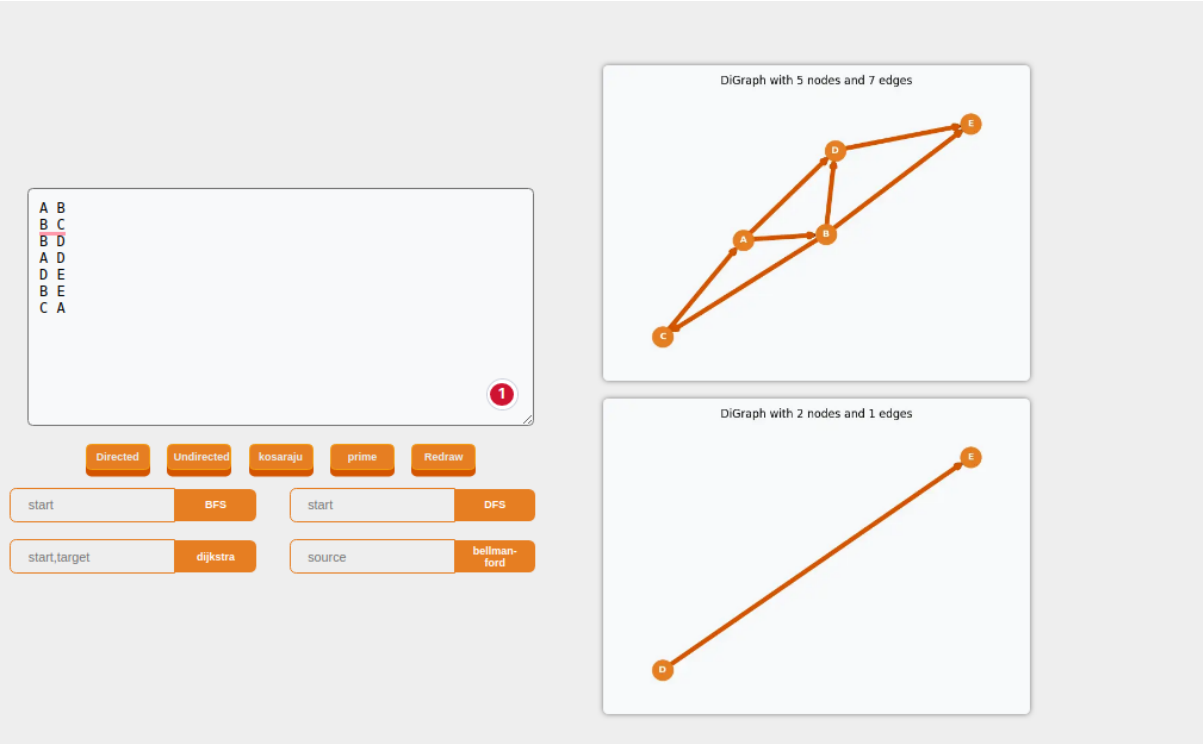
Exemple BFS a partier de A



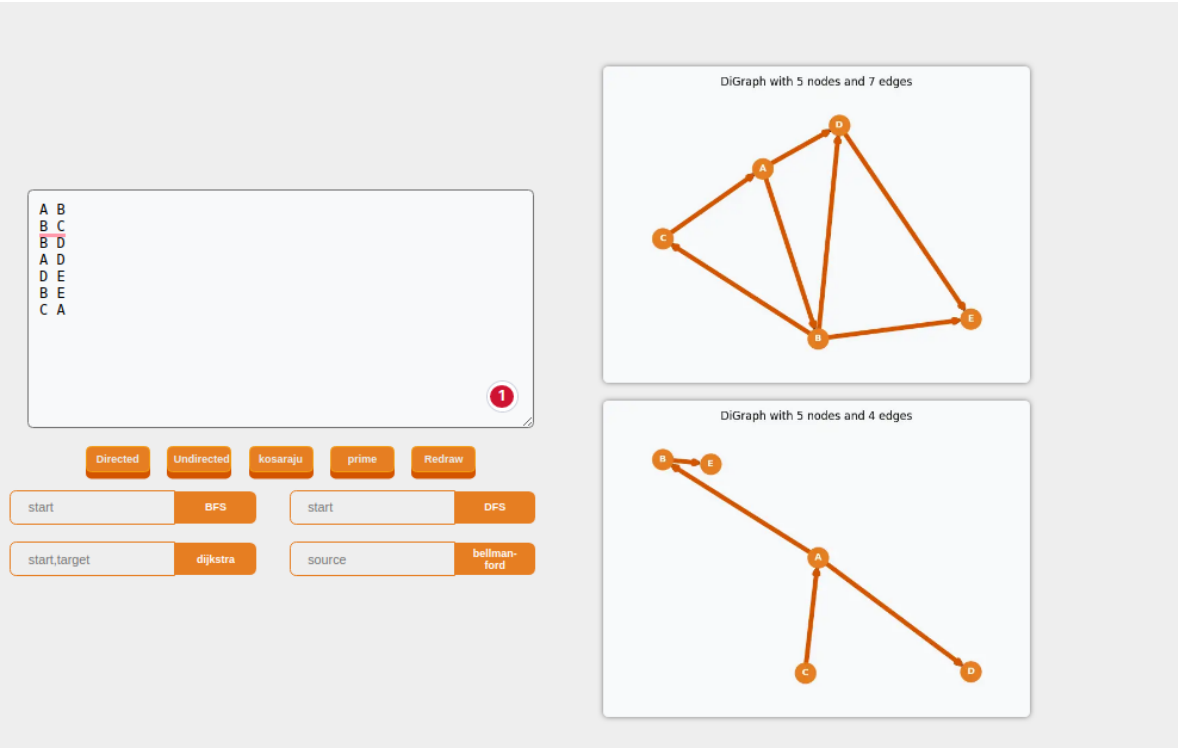
Exemple BFS a partier de E



Exemple DFS a partier de D



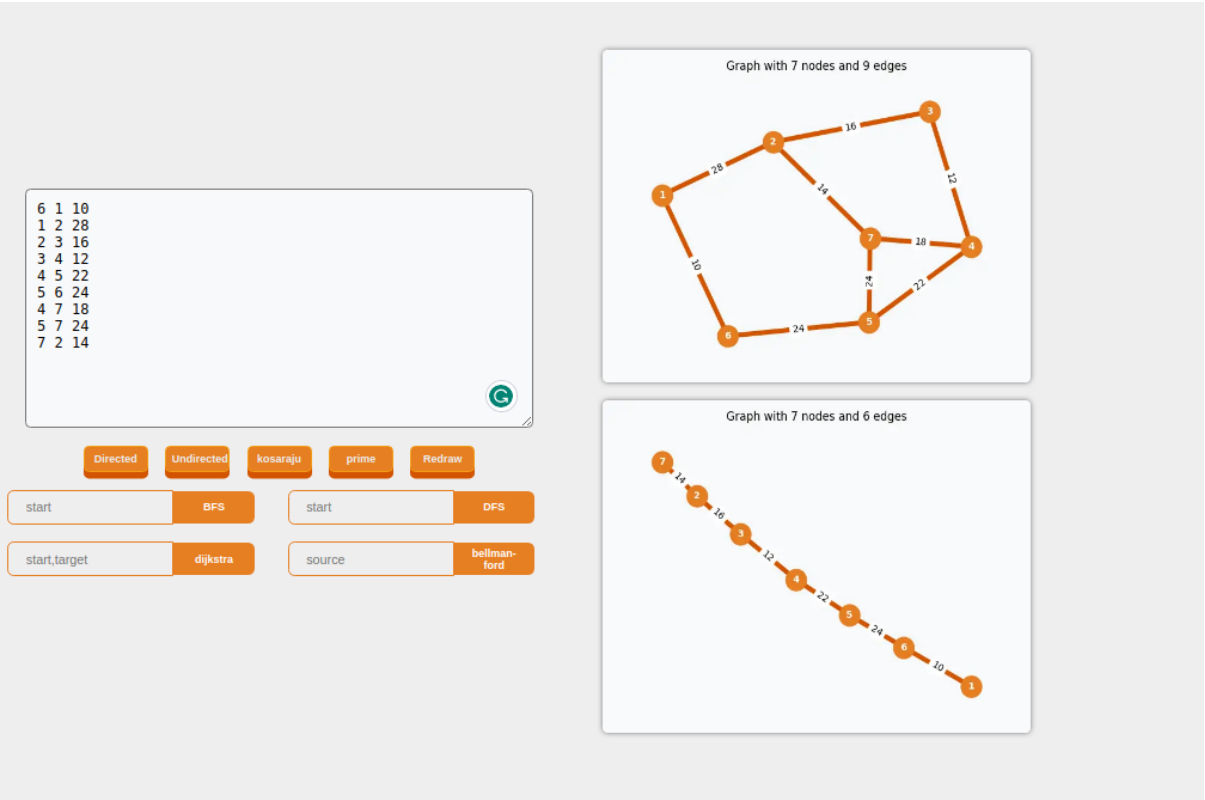
Exemple DFS a partier de C



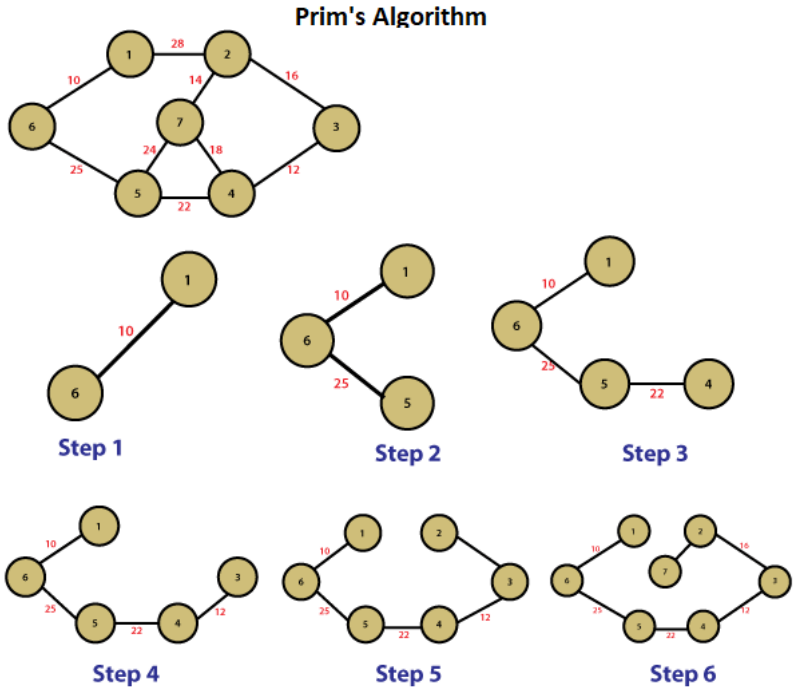
Example de l’algorithme de prime:

pronouns par exemple le graph suivante: (graph simple not orienté pondéré)

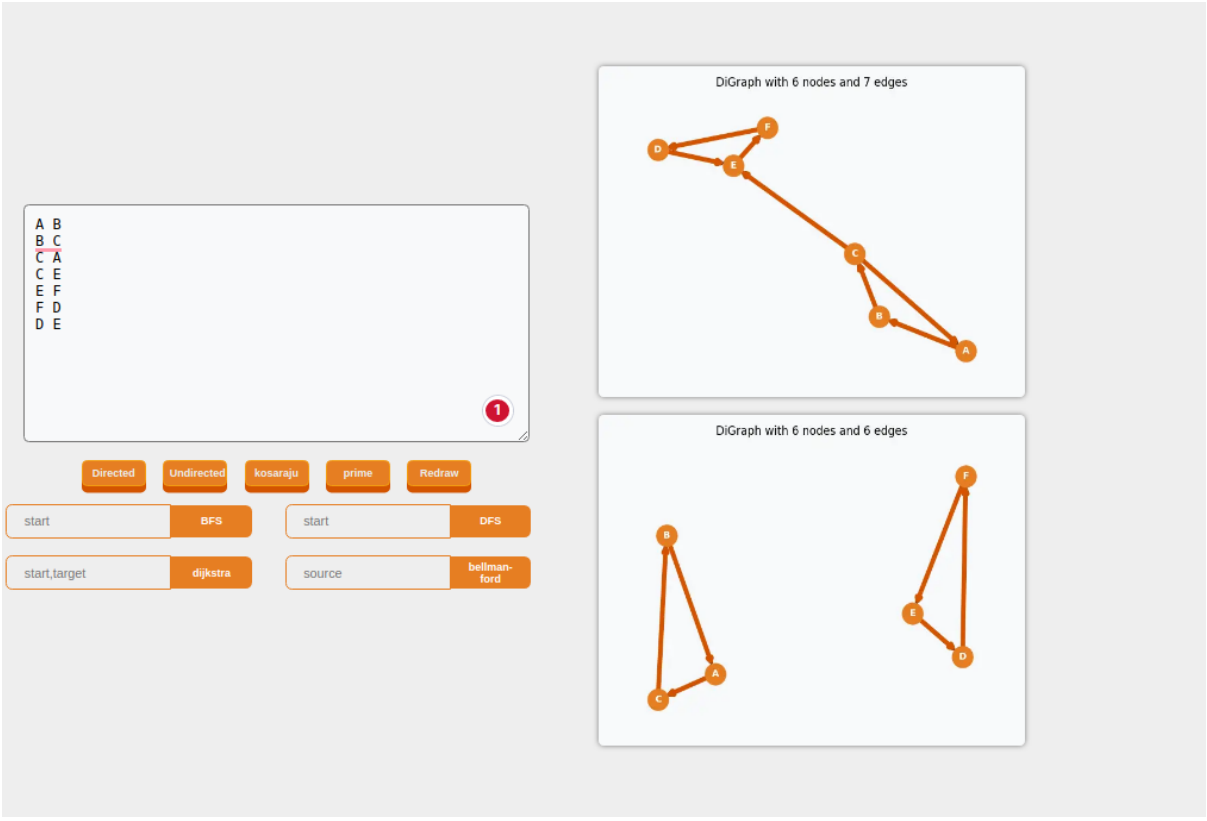
```
6 1 10
1 2 28
2 3 16
3 4 12
4 5 22
5 6 24
4 7 18
5 7 24
7 2 14
```



Explication: en arrière-plan, l'algorithme fait les prochaines étapes



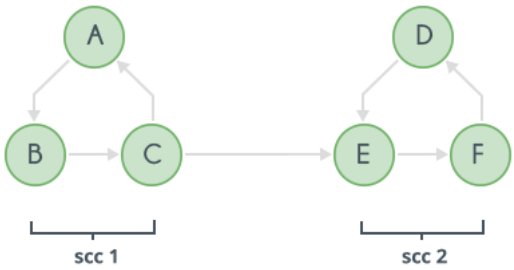
Example de l'algorithme de kosarajus:



Explication:

Example

Consider the following graph:



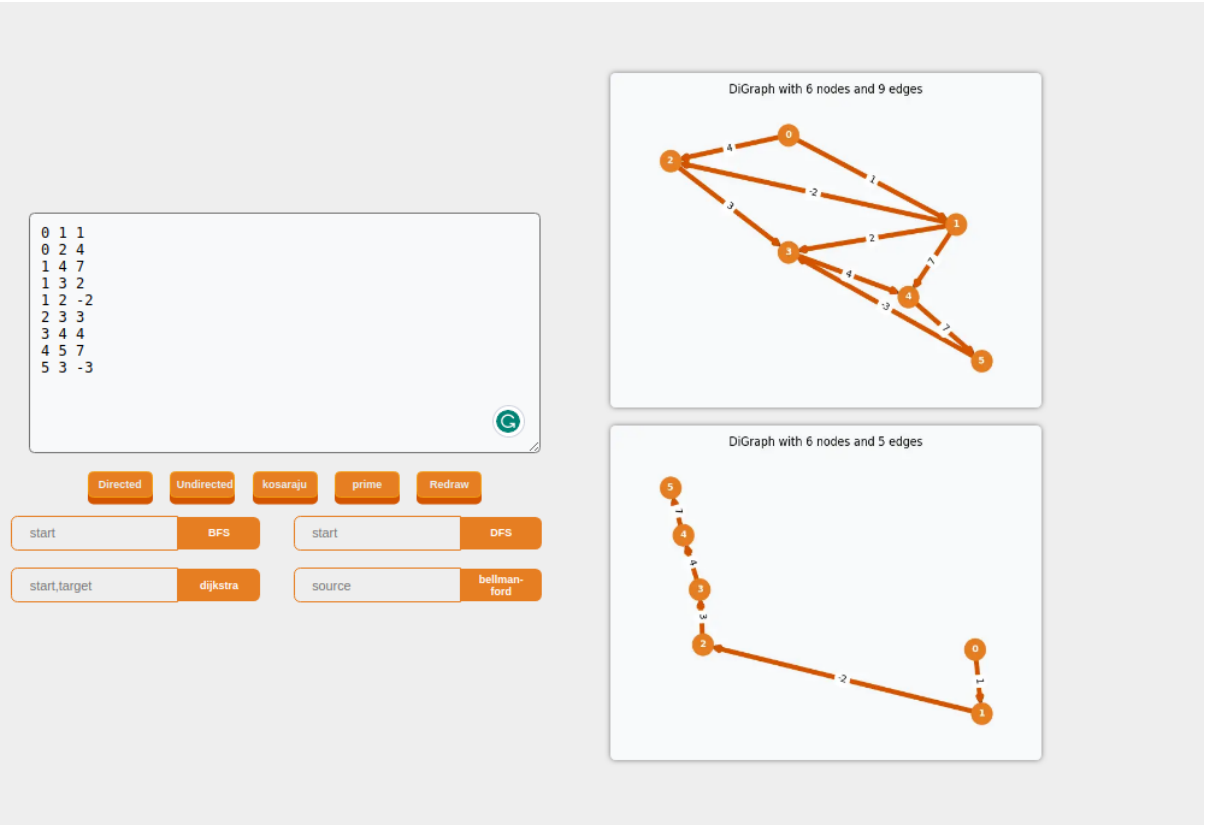
1 It has two strongly connected components scc1 and scc2.

Example de l'algorithme de *bellman Ford:*

Example :

```
0 1 1
0 2 4
1 4 7
1 3 2
1 2 -2
2 3 3
3 4 4
4 5 7
5 3 -3
```

source=0



Explication: