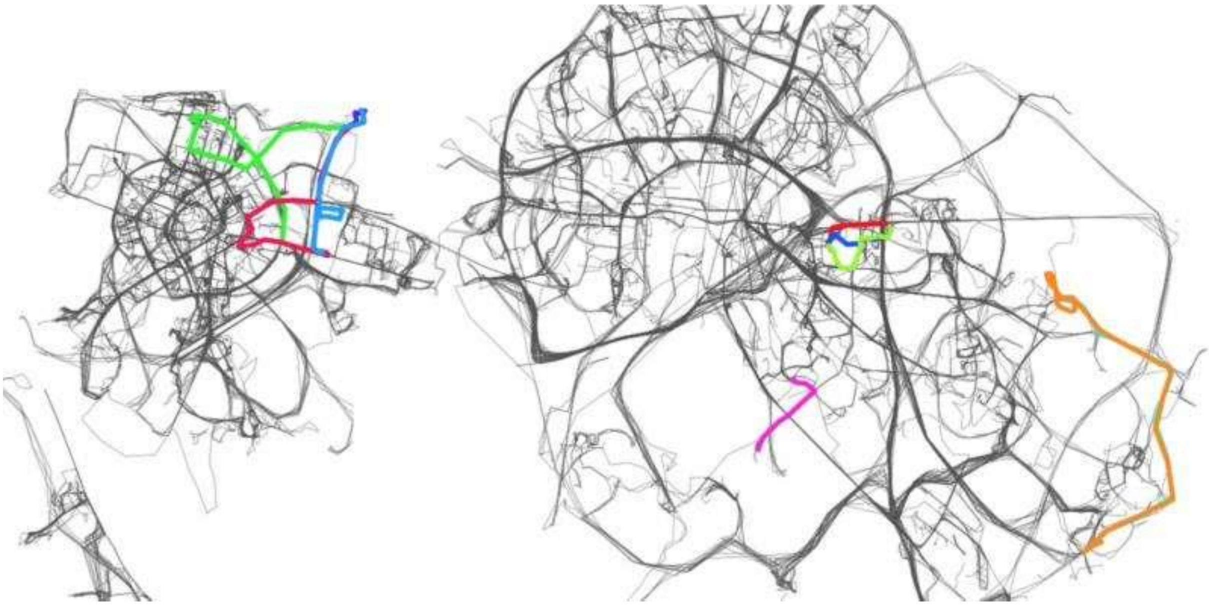
ROYAUME DU MAROC
UNIVERSITÉ ABDELMALEK ESSAÂDI
FACULTÉ DES SCIENCES ET TECHNIQUES
TANGER

CI LSI 2021-2022

# Projet
# Théorie Des Graphes

(implémentation des algorithmes)



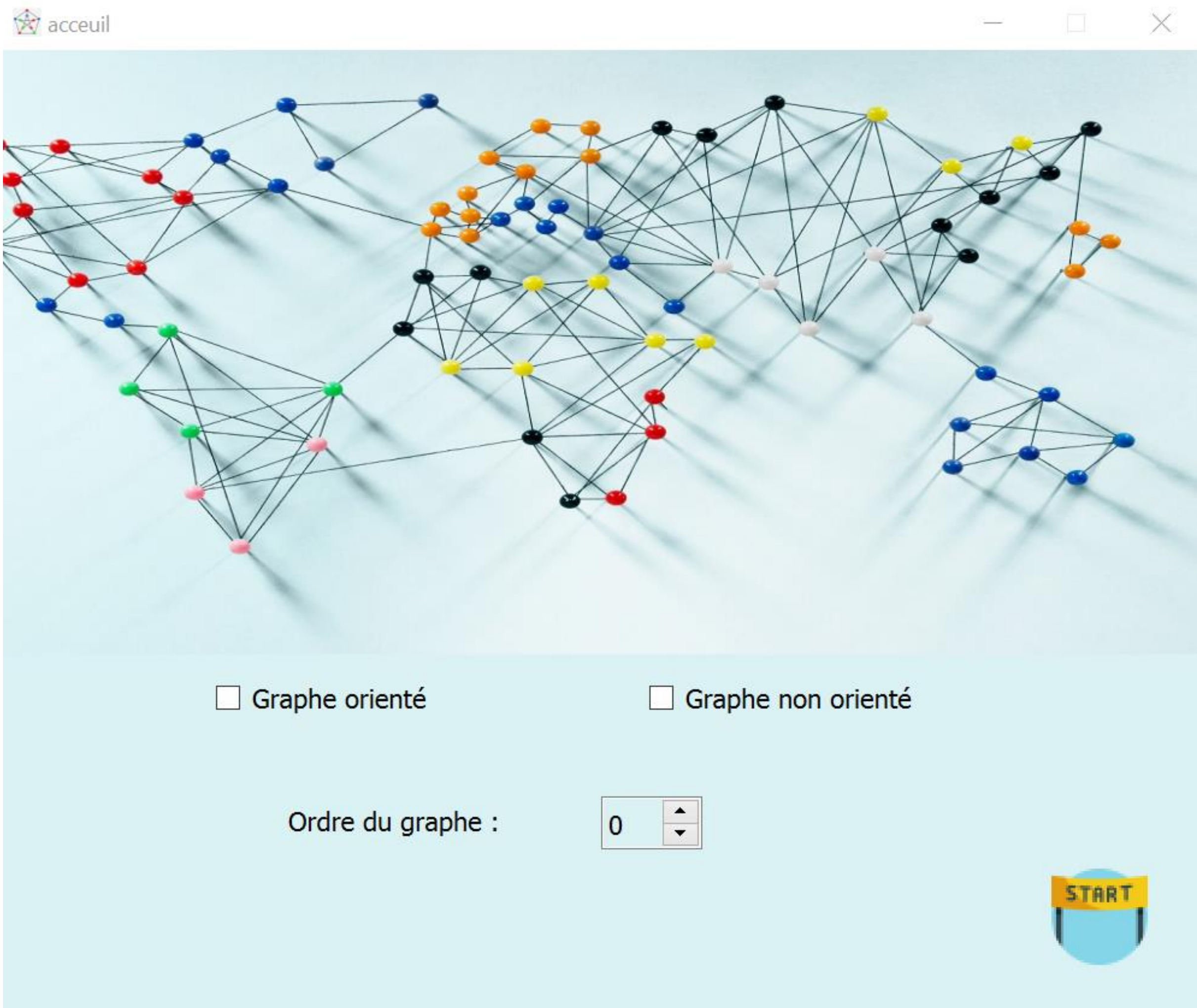
**Encadré par :**

- Mr.GHADI Abderrahim

**Réalisé par :**

- EL KRISSI Achraf
- AKDI Nafia

## 1) ***PAGE D’ACCEUIL :***

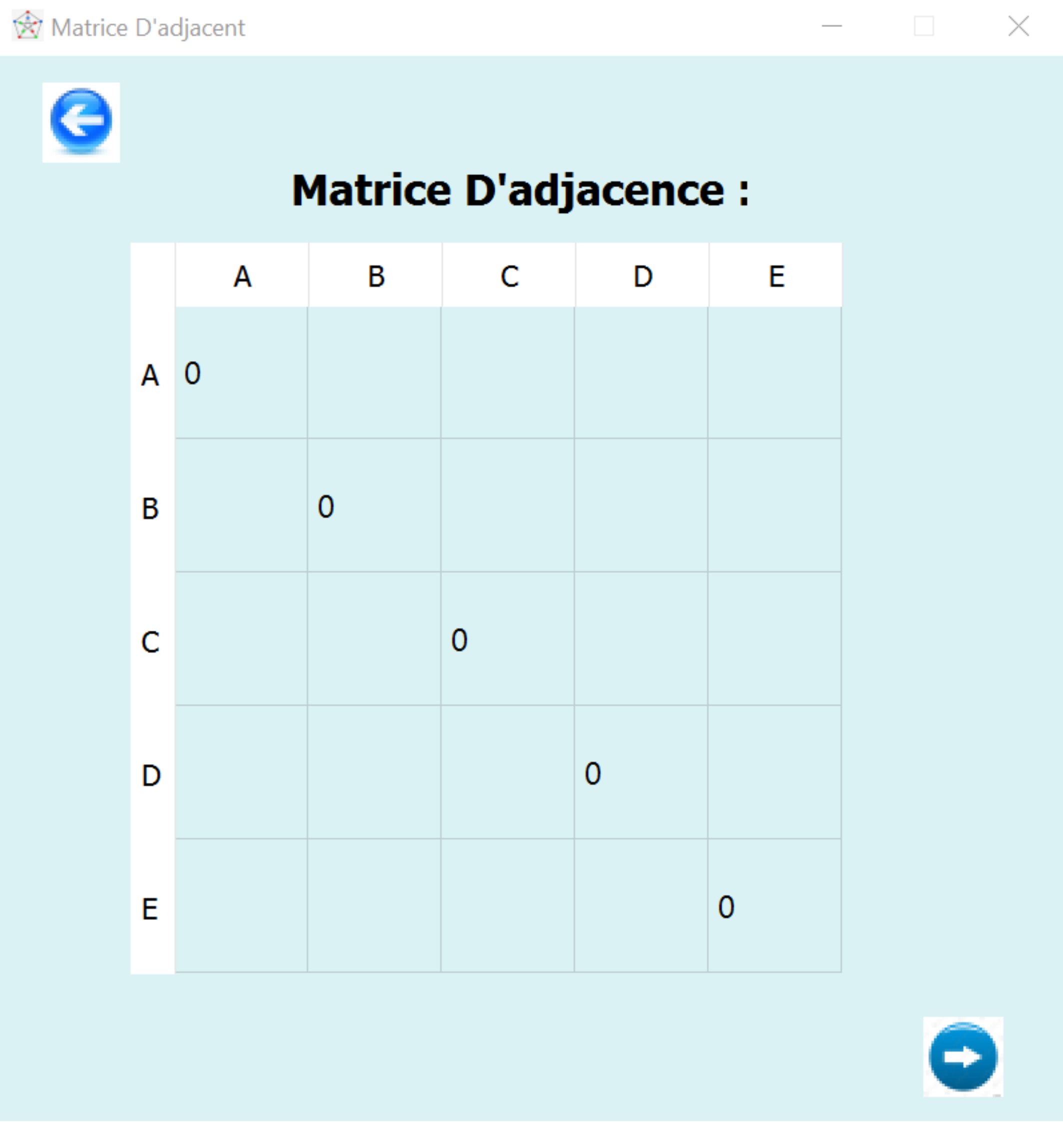



- Dans cette page l’utilisateur doit choisir le type du graphe et préciser son orde ( nombres de sommets ) .

## 2) ***REMPLIR LA MATRICE :***

- Dans cette page l’utilisateur va remplir la matrice carrée d’adjacence de dimension (ordre * ordre) selon l’ordre entré par l’utilisateur dans la page d’acceuil .

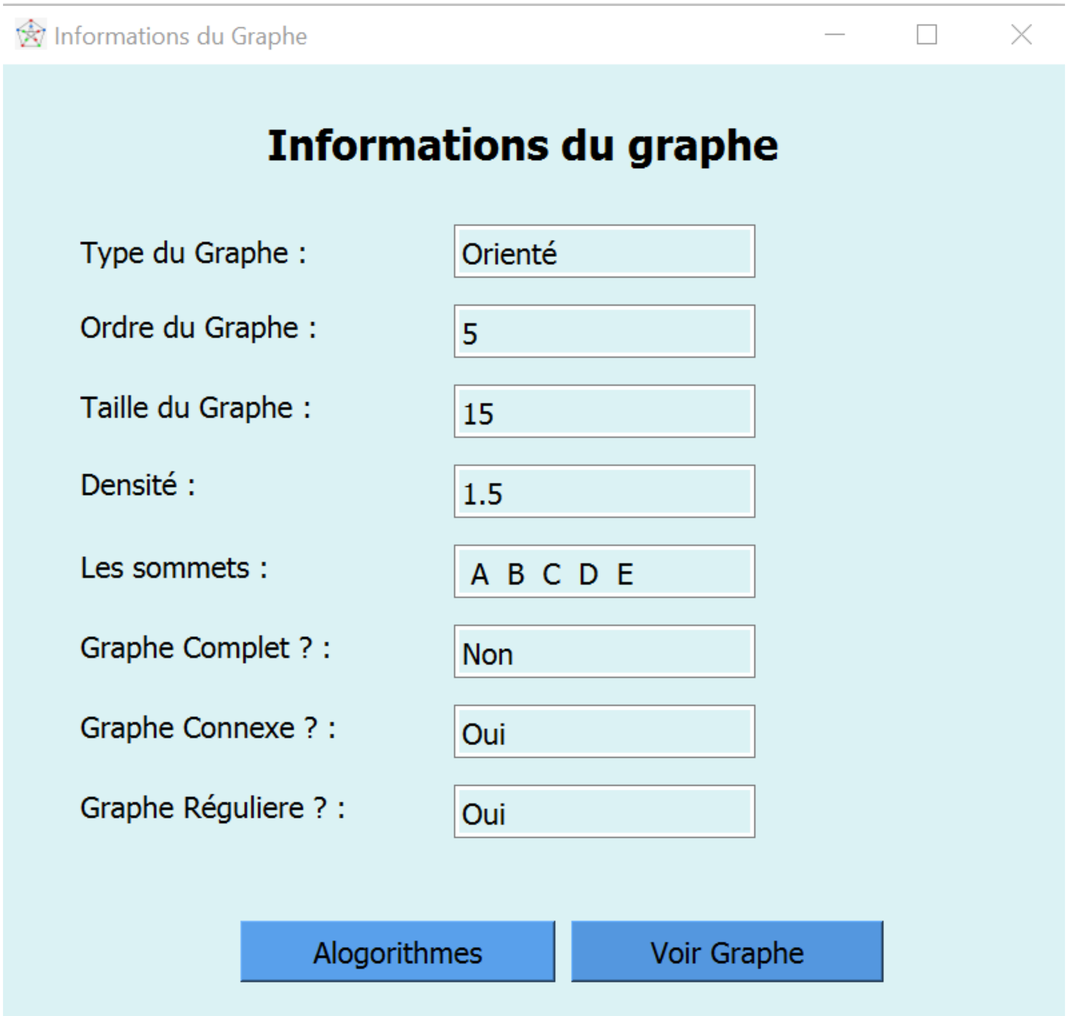
- La diagonale principale de la matrice est par défaut nulle car on va pas  utiliser les boucles .




## 3) ***INFORMATIONS DU GRAPHE :***

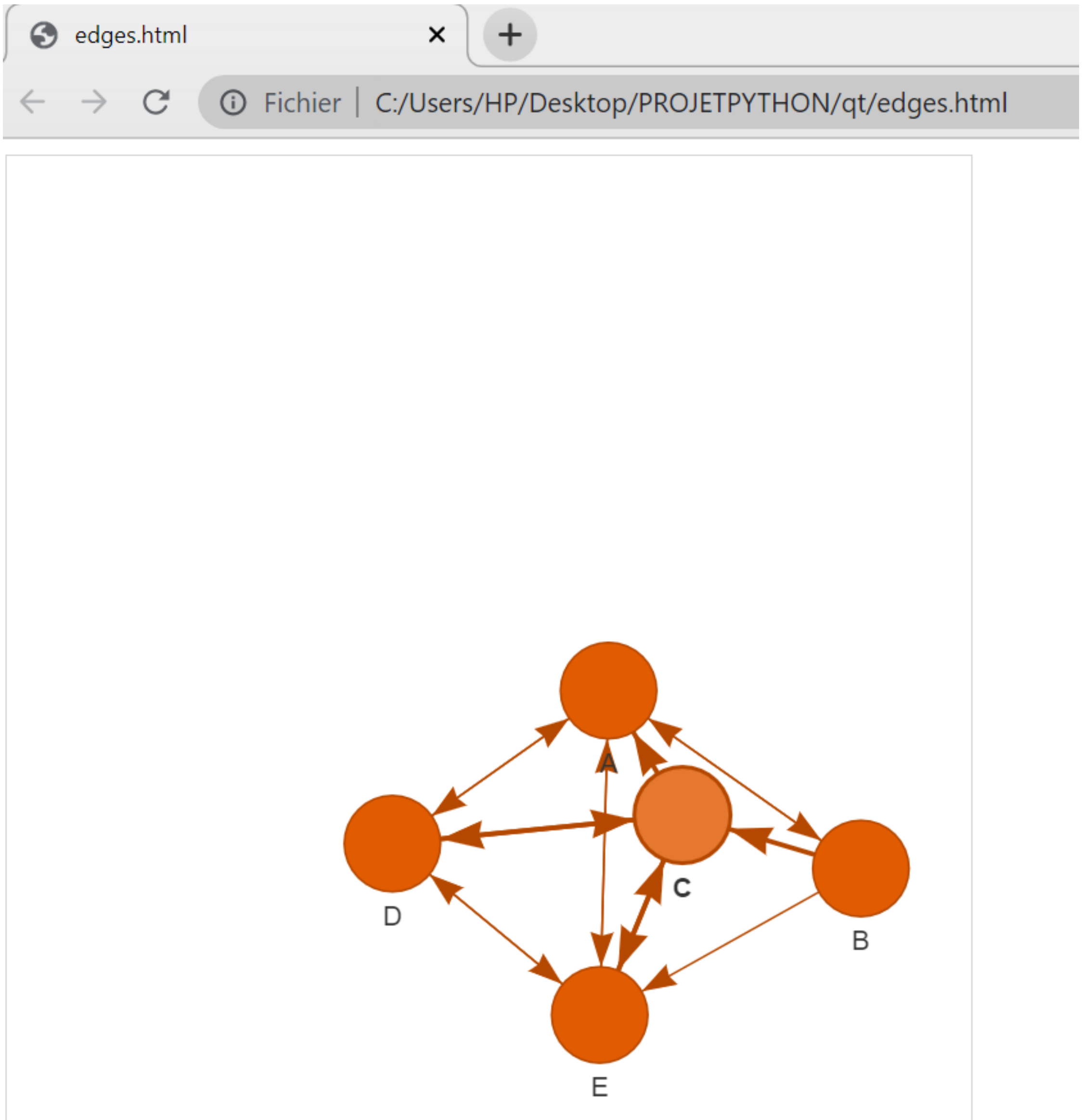
- Lorsque l'utilisateur a rempli la matrice on va l'afficher les informations importantes du graphe :

- Dans la meme page l'utilisateur peut voir le graphe en cliquant sur le button correspondant ou bien acceder a la page consacrée pour les algorithmes .




## 4) ***AFFICHAGE DU GRAPHE :***

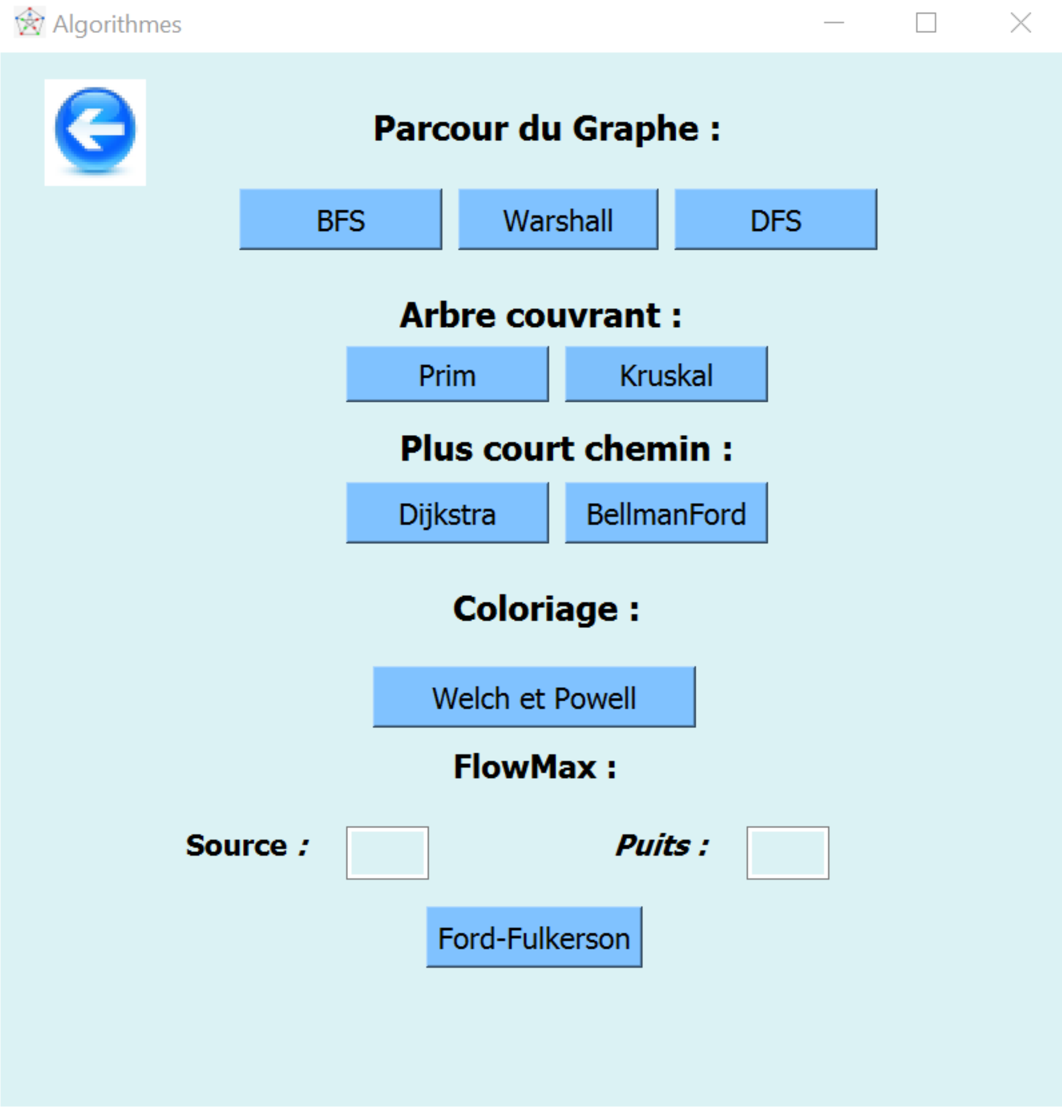
- Lorsque l'utilisateur clique sur le button voir graphe on va le generer une page html pour l'affichage :




## 5) ***ALGORTHMES :***

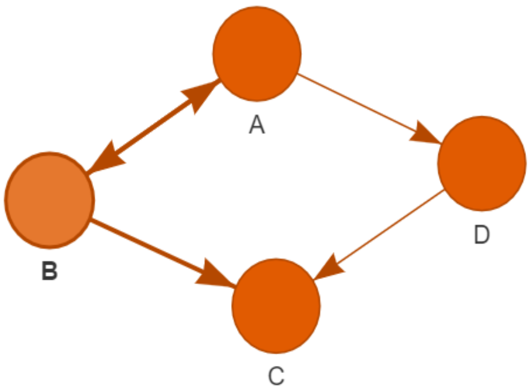
- Lorsque l'utilisateur clique sur Algorithmes , il va passer à la pages des algorithmes .

- Notre page contient les 9 algorithmes :
  - BFS
  - DFS
  - WARSHALL
  - PRIM
  - KRUSKAL
  - DIJIKSTRA
  - BELLMANFORD
  - WELCH ET POWELL
  - FORD-FULKERSON(FLOW-MAX)

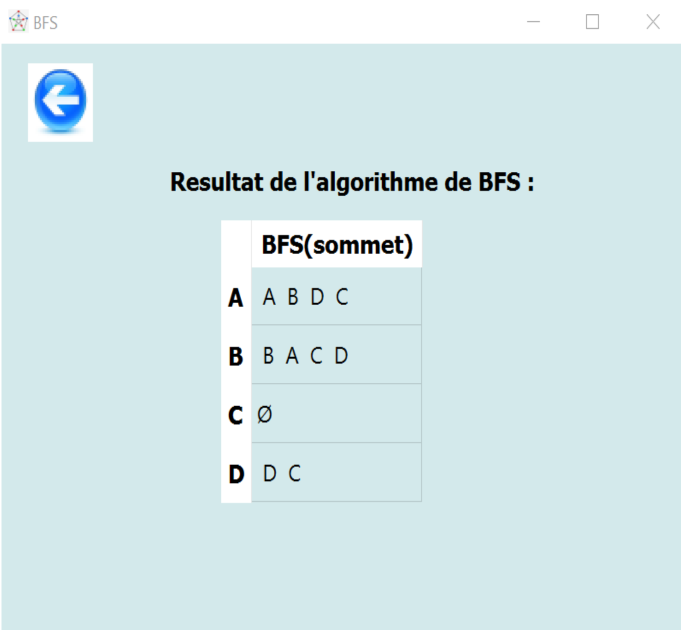


Algorithmes
Parcour du Graphe :
BFS
Warshall
DFS
Arbre couvrant :
Prim
Kruskal
Plus court chemin :
Dijkstra
BellmanFord
Coloriage :
Welch et Powell
FlowMax :
Source :
Puits :
Ford-Fulkerson

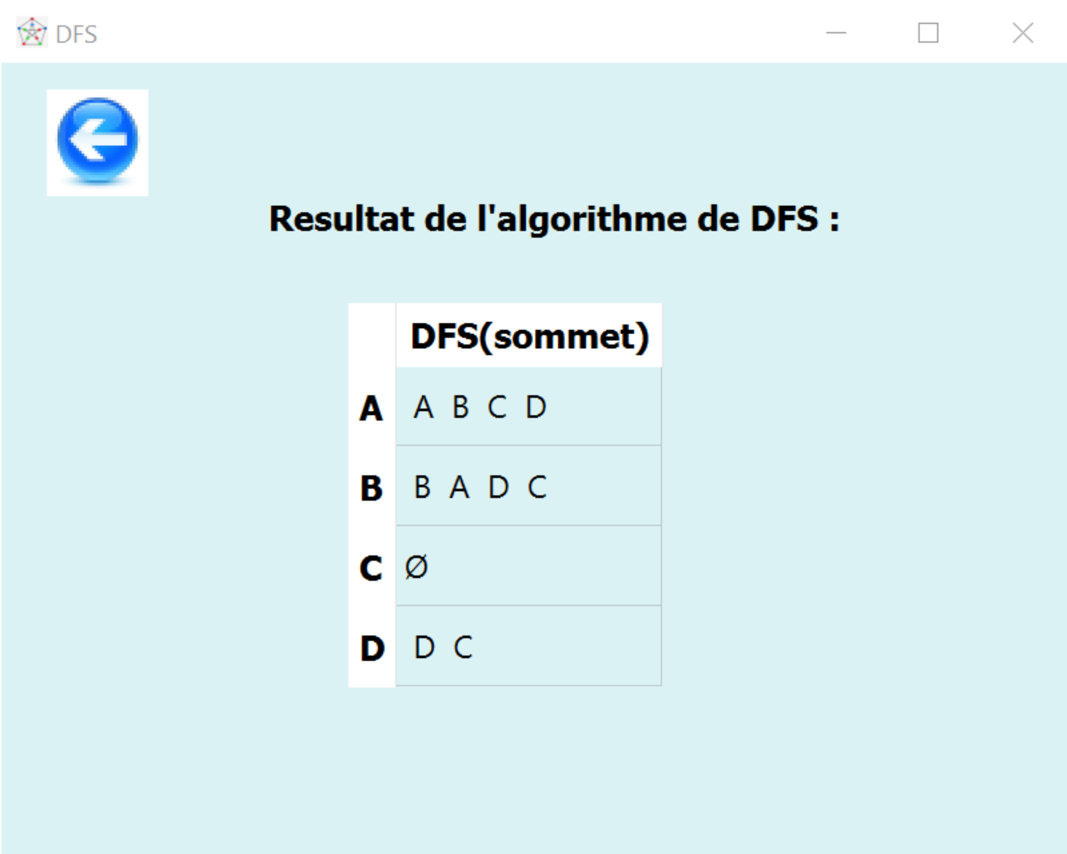

## 5-1) BFS :



A
B
C
D

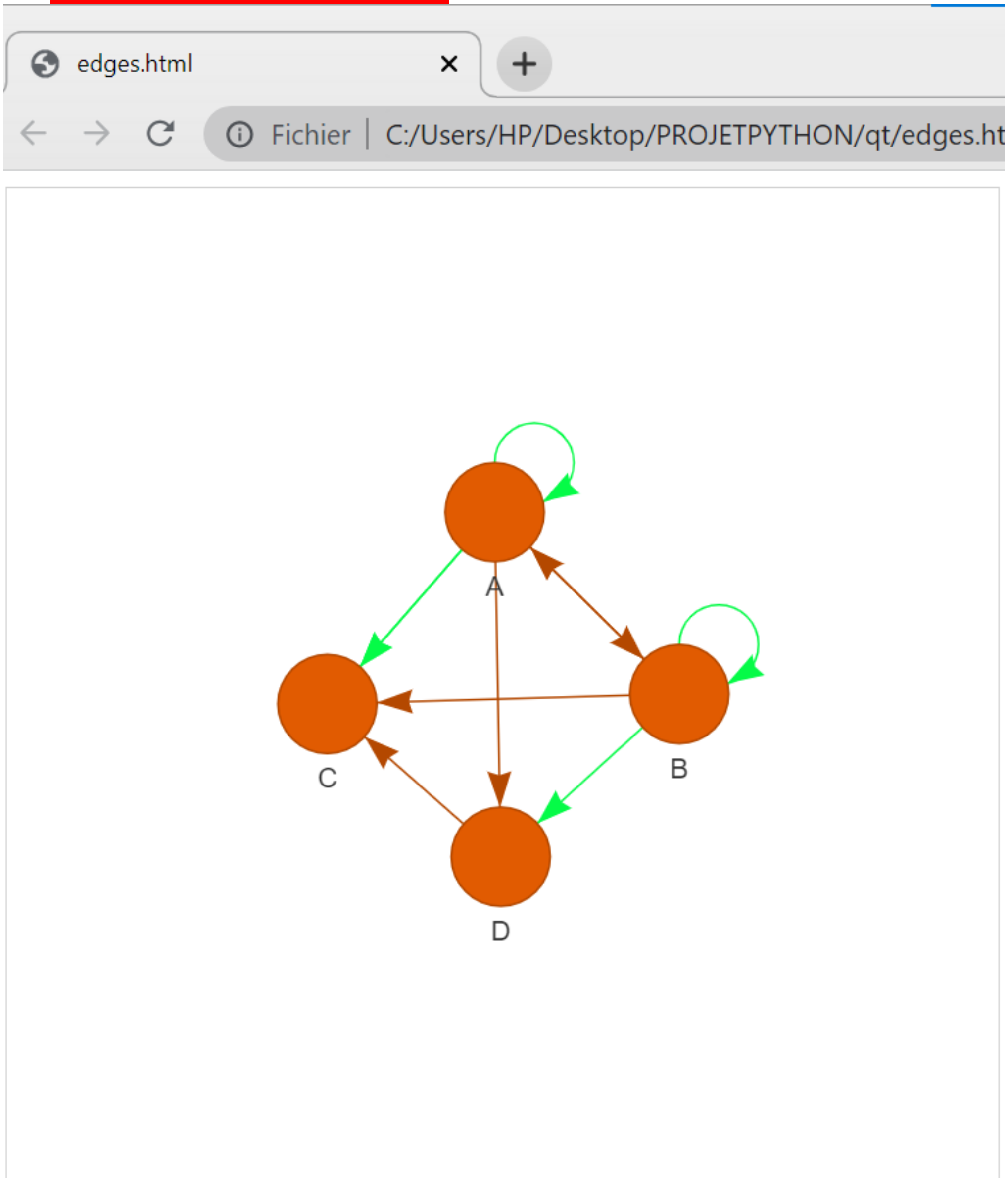



## 5-2) DFS :




- **ON PEUT VOIR CLAIREMENT LA DIFFERNCE ENTRE BFS(B) ET DFS(B) .**

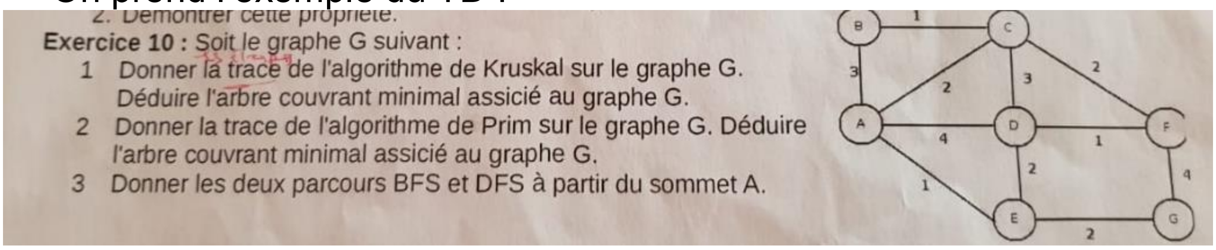
## - 5-3) WARSHALL :




- Les arcs en vert sont ajoutés après l'application de Warshall.

## - 5-4) KRUSKAL :

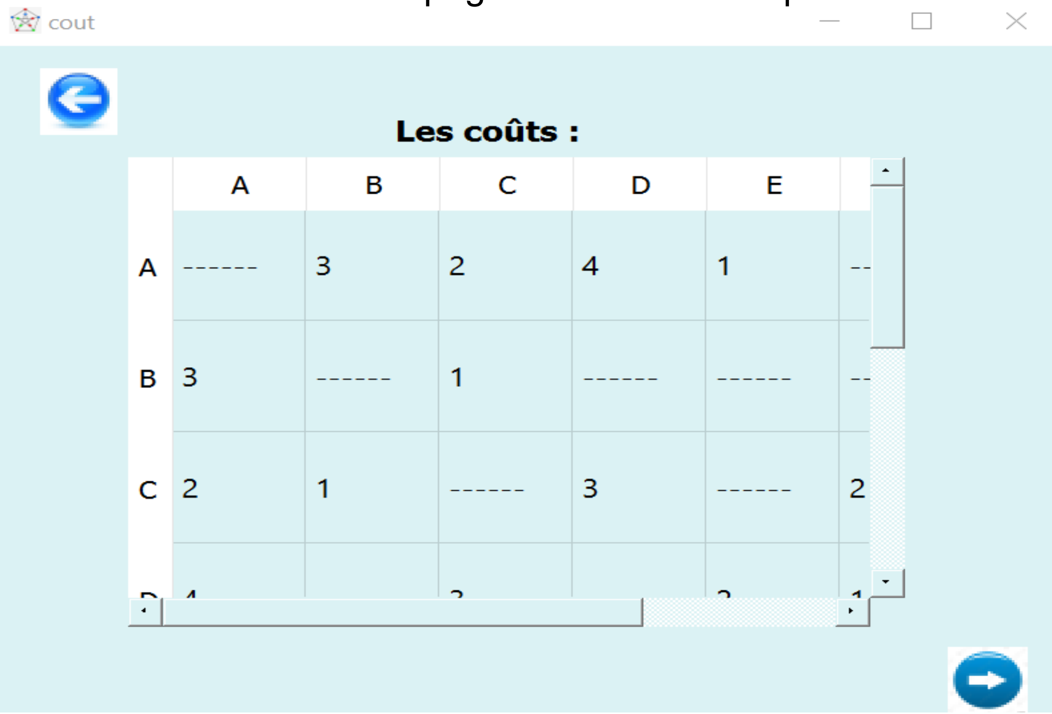
- On prend l'exemple du TD :


2. Démontrer cette propriété.

**Exercice 10 :** Soit le graphe G suivant :

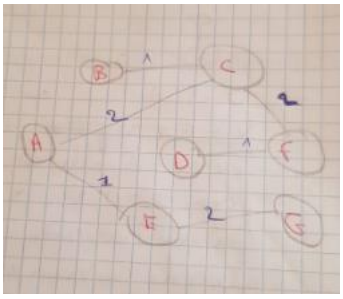
1. Donner la trace de l'algorithme de Kruskal sur le graphe G. Déduire l'arbre couvrant minimal assicié au graphe G.
2. Donner la trace de l'algorithme de Prim sur le graphe G. Déduire l'arbre couvrant minimal assicié au graphe G.
3. Donner les deux parcours BFS et DFS à partir du sommet A.

- Tout d’abord on a une page des couts a remplir :

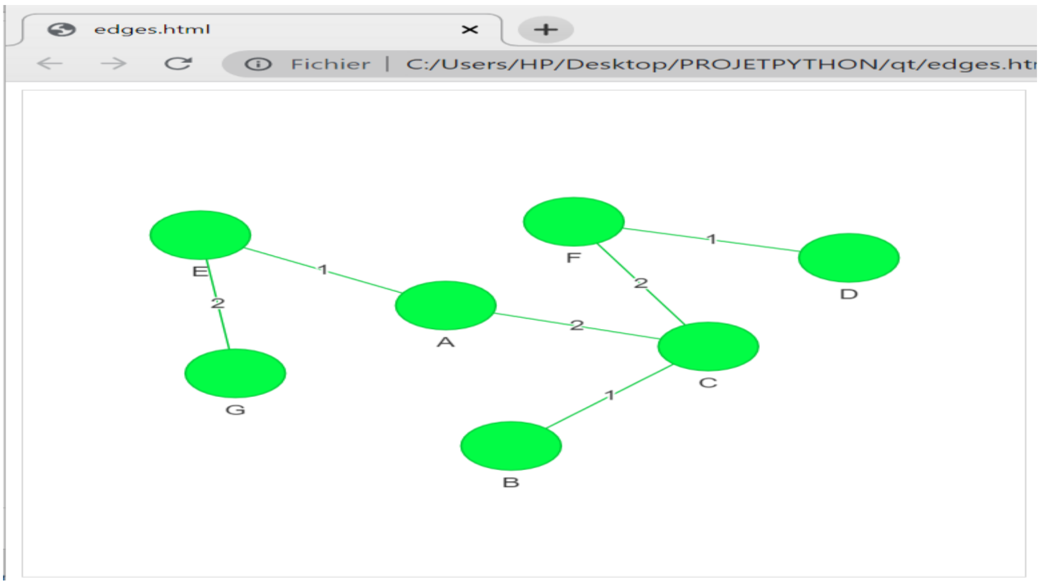



| | A | B | C | D | E | |
|---|---|---|---|---|---|---|
| A | ------ | 3 | 2 | 4 | 1 | -- |
| B | 3 | ------ | 1 | ------ | ------ | -- |
| C | 2 | 1 | ------ | 3 | ------ | 2 |

- **Resultat attendu :**

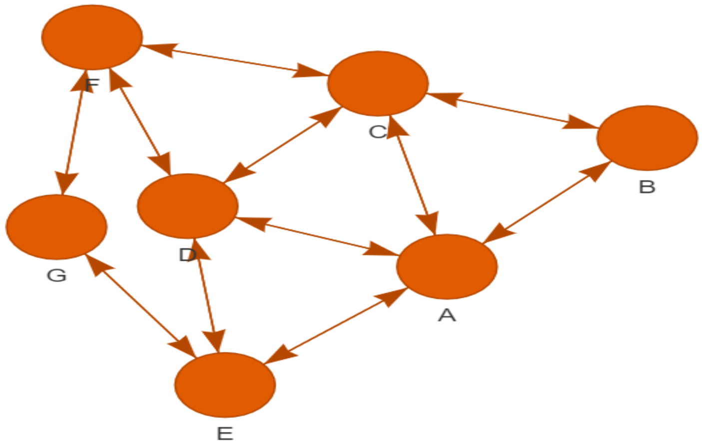


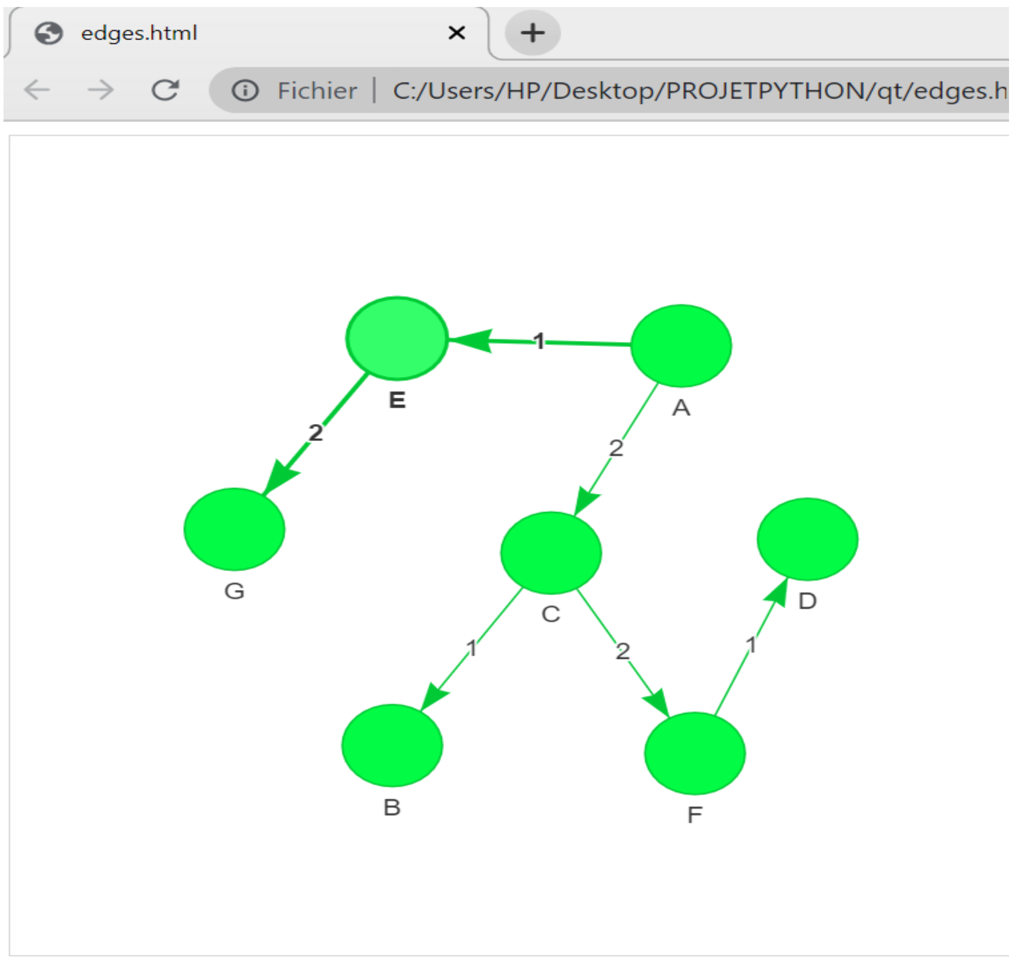
- **Resultat KRUSKAL:**



## - 5-5) PRIM :

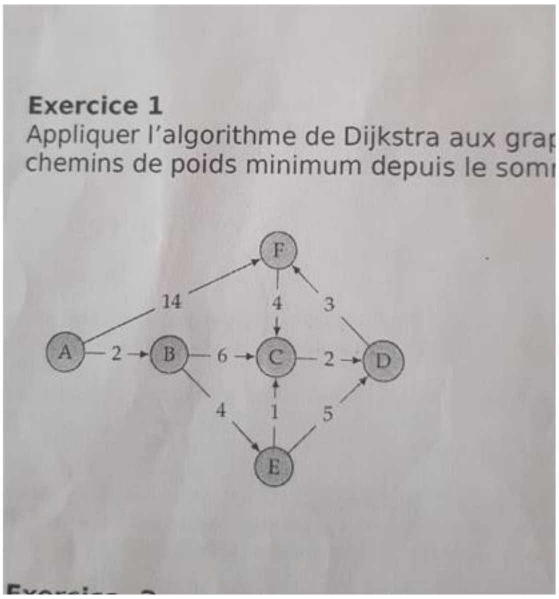
- On prend meme exemple du Td mais cette fois on va le supposer **orienté**:

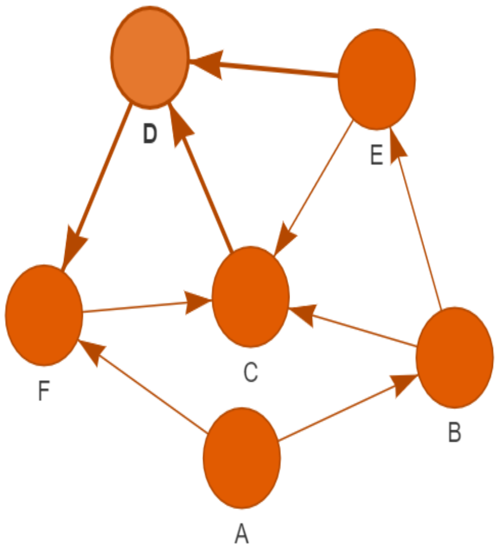


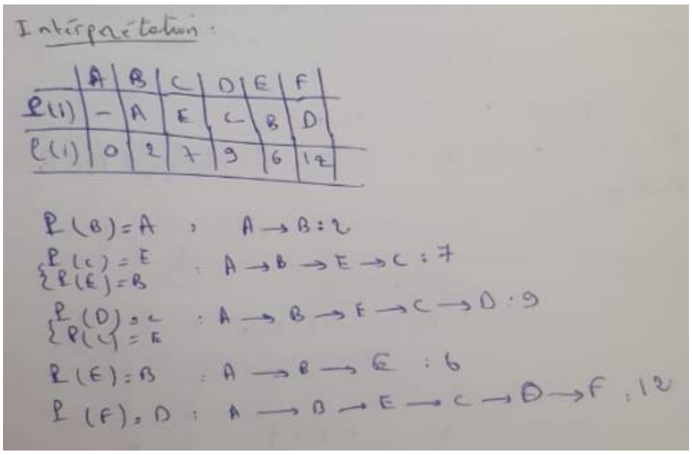
## - *5-6) DIJIKSTRA:*

- On prend l'exemple du TD :




- Resultat attendu pour le sommet du depart A :



Interprétation :

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| P(i) | – | A | E | C | B | D |
| P(i) | 0 | 2 | 7 | 9 | 6 | 12 |

P(B) = A ; A → B : 2

{P(C) = E
{P(E) = B : A → B → E → C : 7

{P(D) = C
{P(C) = E : A → B → E → C → D : 9

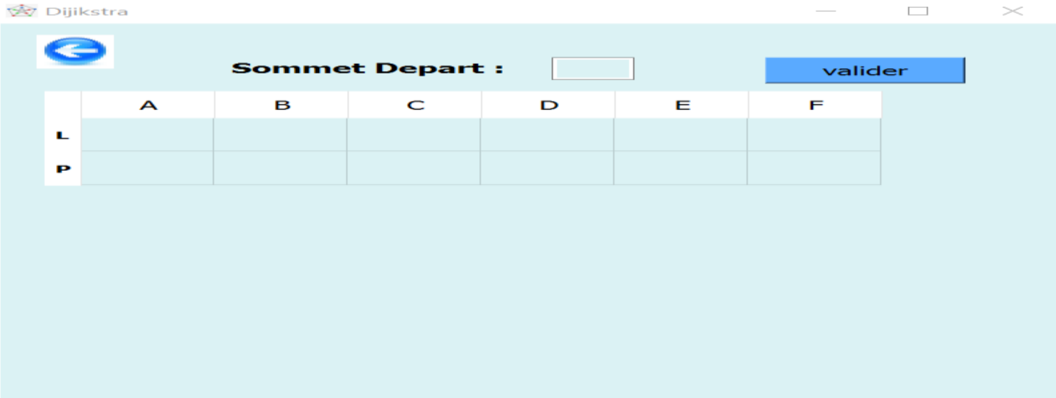
P(E) = B : A → B → E : 6

P(F) = D : A → B → E → C → D → F : 12

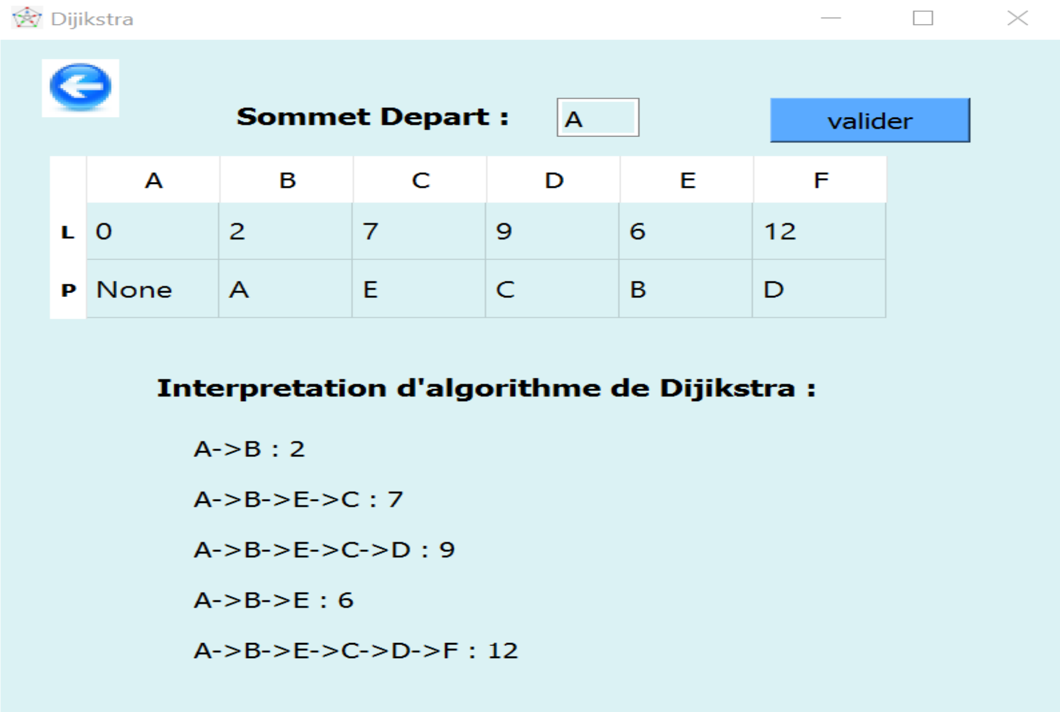
- On remplit les couts :

- Notre resultat :

**Avant qu’on a choisi le sommet de depart :**

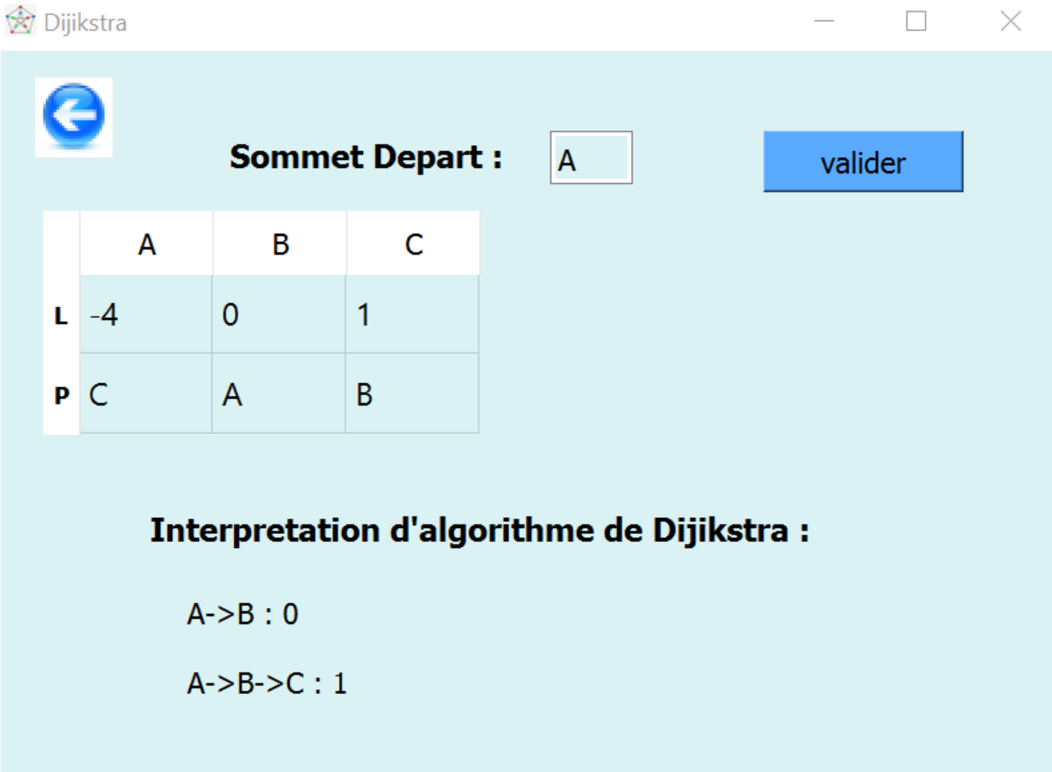



**Apres qu’on a choisi le sommet de depart :**




- **Testons maintenant sur un graphe contient un circuit absorbant :**

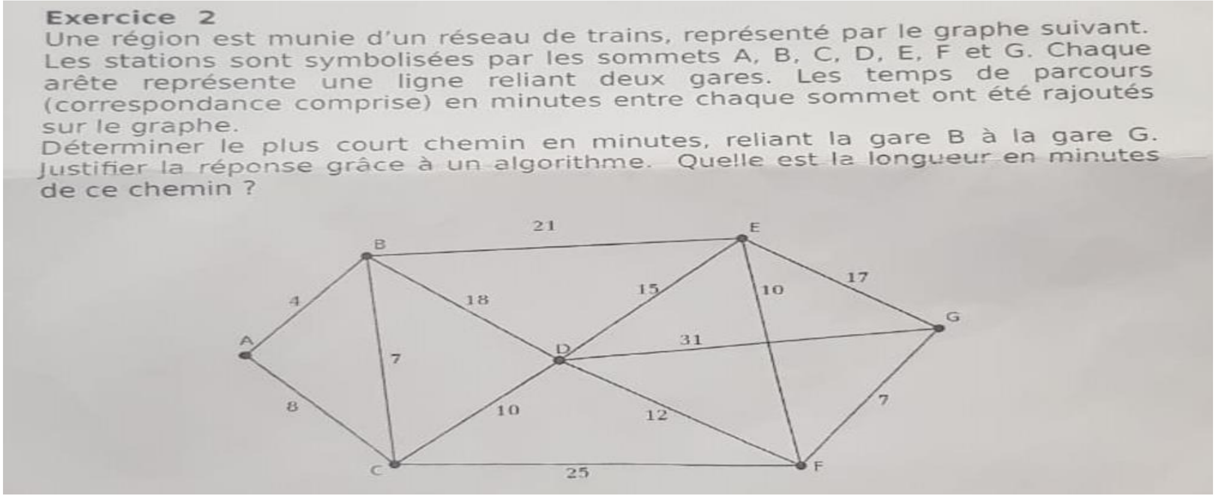

- **Resultat :**




- Les resultats sont incompatibles car l'algorithme de Dijikstra ne prend pas en consideration des circuit absorbant (n'est pas comme bellman ford ).

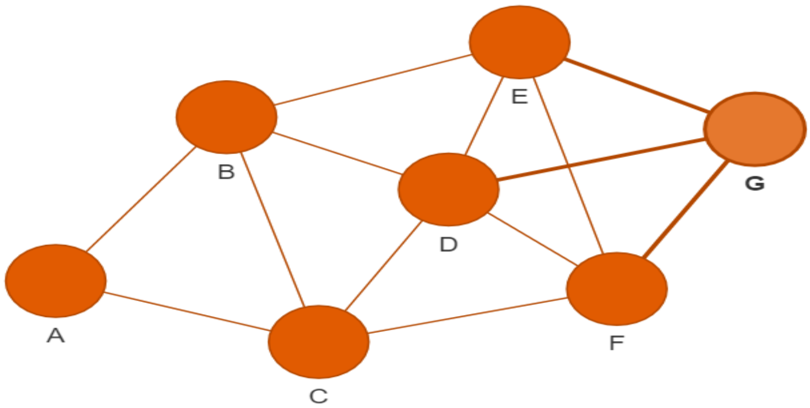
## - 5-7) BELLMANFORD:

- On prend l'exemple du TD suivant :

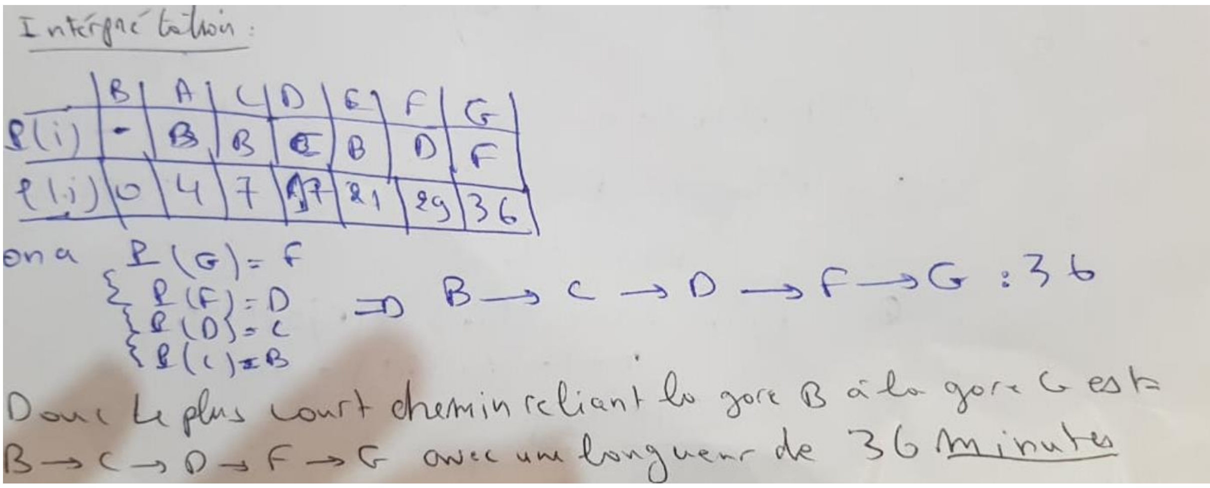


**Exercice 2**
Une région est munie d'un réseau de trains, représenté par le graphe suivant. Les stations sont symbolisées par les sommets A, B, C, D, E, F et G. Chaque arête représente une ligne reliant deux gares. Les temps de parcours (correspondance comprise) en minutes entre chaque sommet ont été rajoutés sur le graphe.
Déterminer le plus court chemin en minutes, reliant la gare B à la gare G. Justifier la réponse grâce à un algorithme. Quelle est la longueur en minutes de ce chemin ?






- Rsultat attendu :



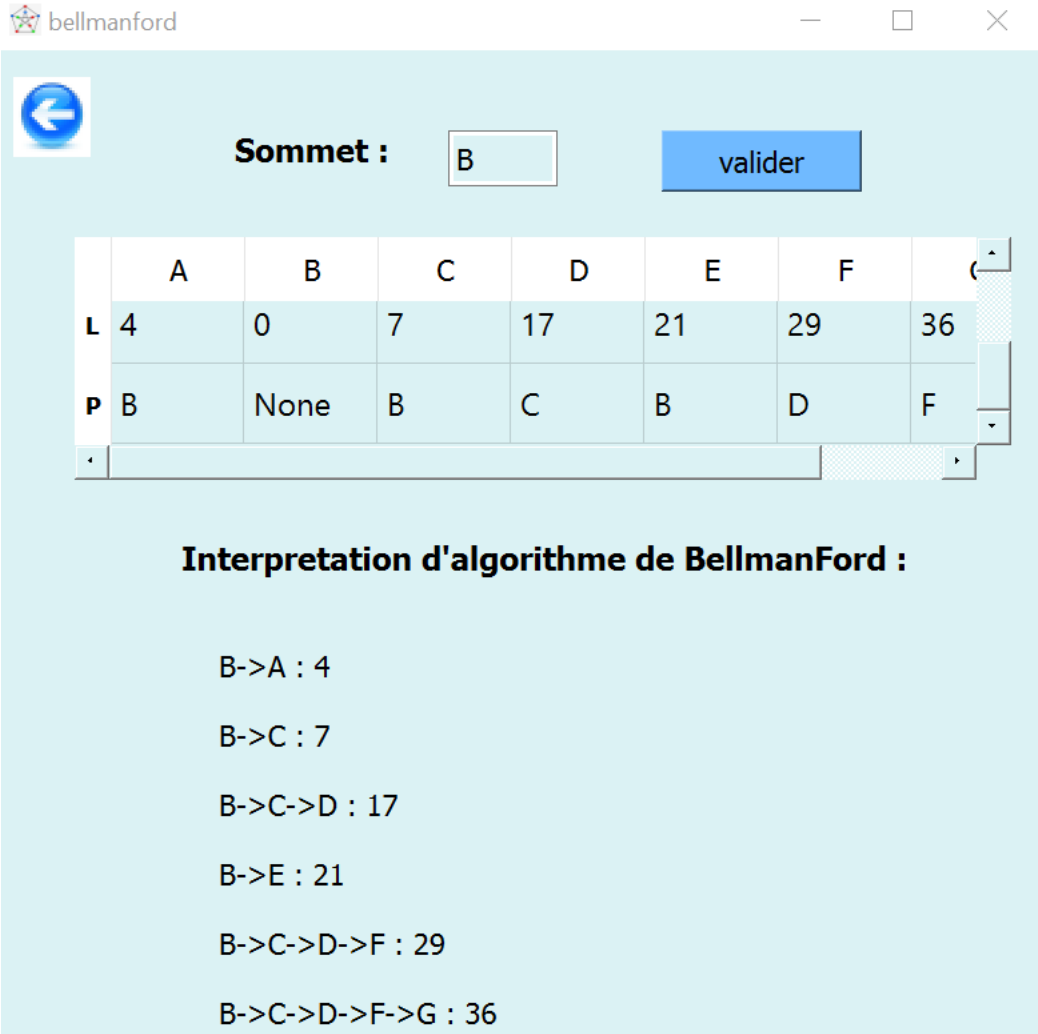
Intérprétation :

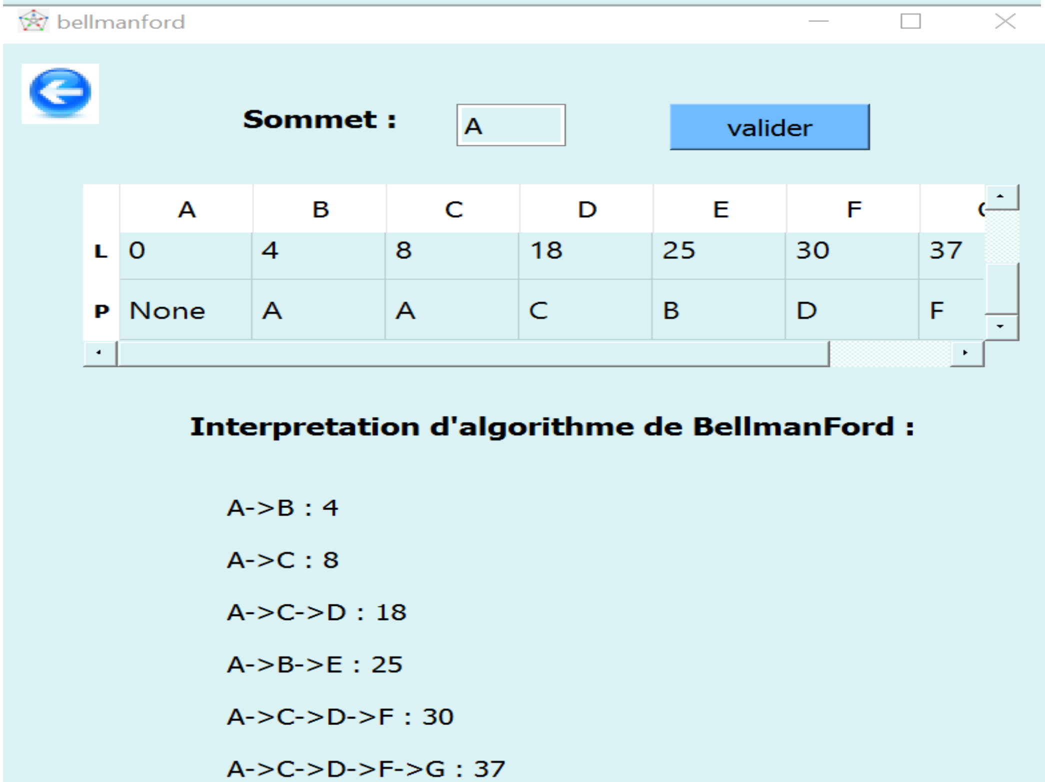
| | B | A | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| P(i) | - | B | B | C | B | D | F |
| ℓ(i) | 0 | 4 | 7 | 17 | 21 | 29 | 36 |

on a
$P(G) = F$
$\{ P(F) = D$
$\{ P(D) = C$
$\{ P(C) = B$
$\Rightarrow B \to C \to D \to F \to G : 36$

Donc le plus court chemin reliant la gare B à la gare G est $B \to C \to D \to F \to G$ avec une longueur de 36 minutes

- Notre solution :

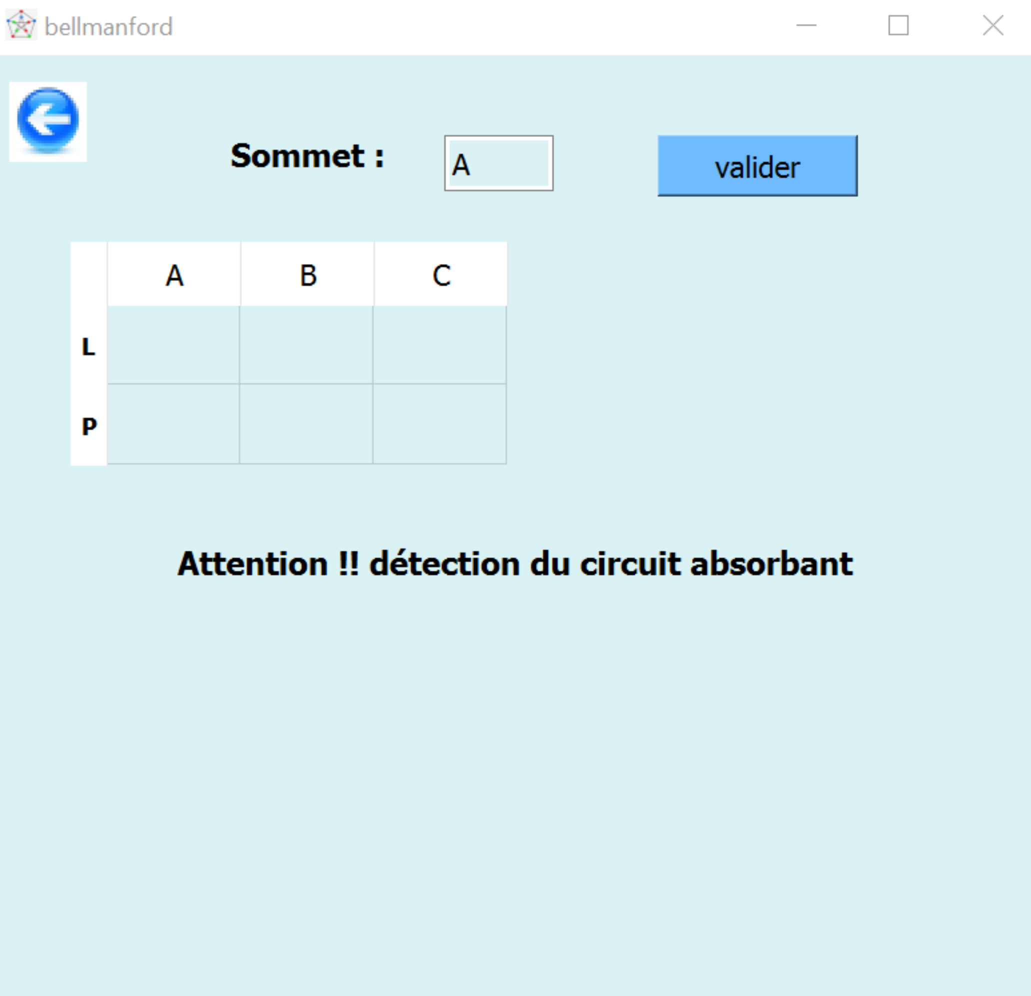

bellmanford
Sommet :
B
valider
A B C D E F
L 4 0 7 17 21 29 36
P B None B C B D F
Interpretation d'algorithme de BellmanFord :
B->A : 4
B->C : 7
B->C->D : 17
B->E : 21
B->C->D->F : 29
B->C->D->F->G : 36


bellmanford
Sommet :
A
valider
A B C D E F
L 0 4 8 18 25 30 37
P None A A C B D F
Interpretation d'algorithme de BellmanFord :
A->B : 4
A->C : 8
A->C->D : 18
A->B->E : 25
A->C->D->F : 30
A->C->D->F->G : 37

- **Testons maintenant sur un graphe contient un circuit absorbant meme exemple de DIJIKSTRA :**

**Les coûts :**

| | **A** | B | C |
|---|---|---|---|
| A | ------ | 2 | ------ |
| B | ------ | ------ | 1 |
| **C** | -5 | ------ | ------ |

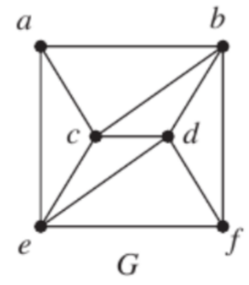


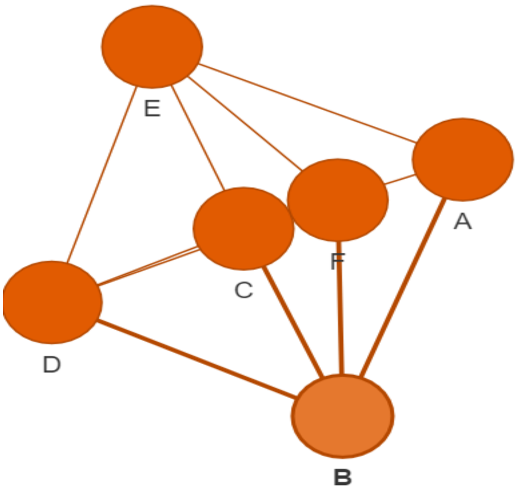

- Il prend à la consideration les circuits absorbants .

- ## 5-8) Coloriage (Welch et Powell):

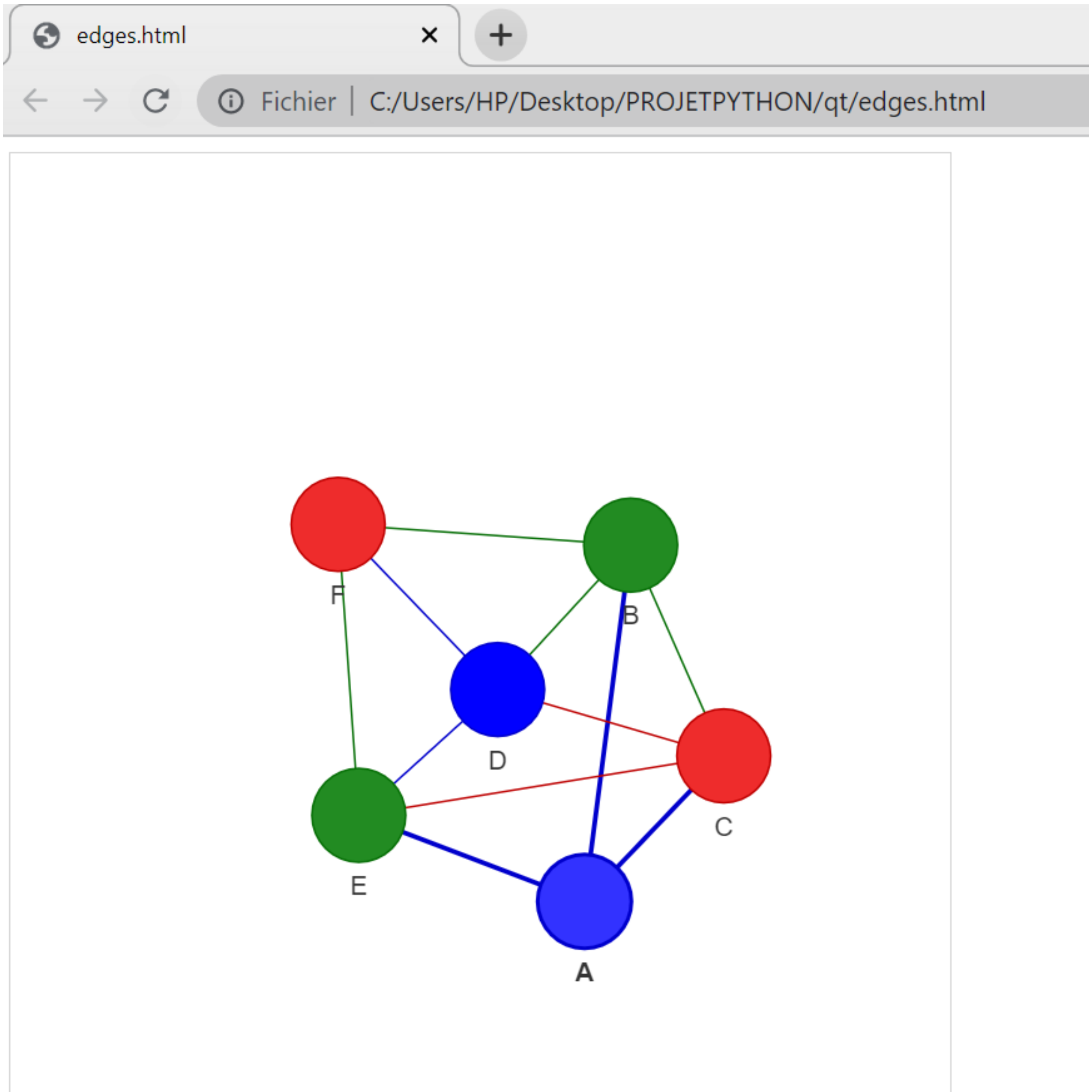
- On prend l'exemple du cour suivant :

**Exemple 1**: Appliquons cet algorithme au graphe ci-dessous :

| | |
|---|---|
| • Liste ordonnée : {b, c, d, e, a, f}<br>1ère couleur : pour les sommets b et e.<br>• Liste ordonnée : {c, d, a, f}<br>2ème couleur : pour les sommets c et f.<br>• Liste ordonnée : {d, a}<br>3ème couleur : pour les sommets d et a. |  |




- Notre Resultat :

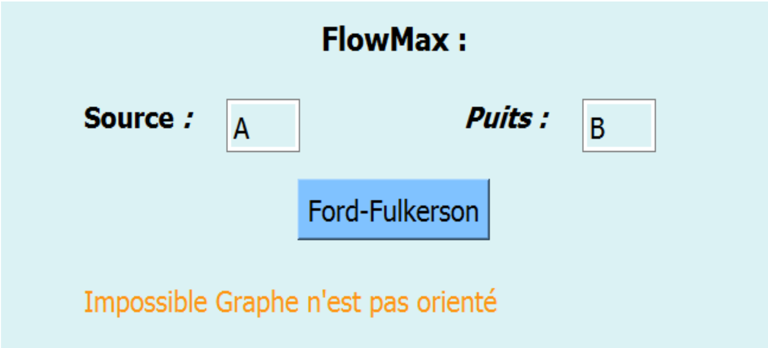


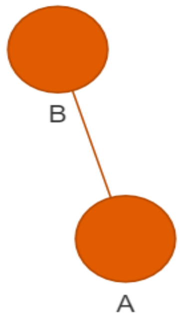

## - ***5-8) FlowMax (Ford-Fulkerson):***

- On sait que l'algorithme de Ford-Fulkerson  est applicable sur les réseaux  de transport c-à-d :

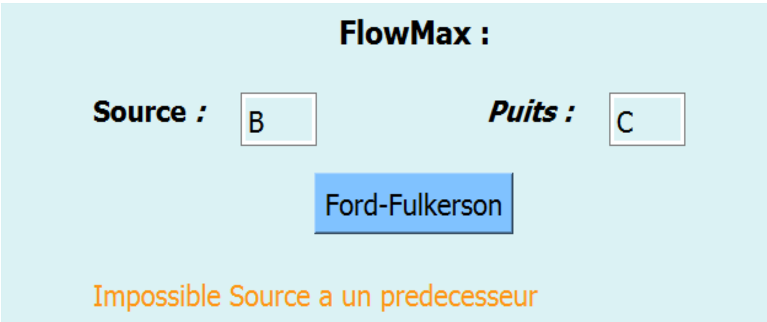
    - Le graphe doit etre **orienté**
    - La source **S**  doit etre sans prédécesseur
    - Le puits  **P**  doit etre sans successeur
    - Au mois un chemin entre **S** et **P**

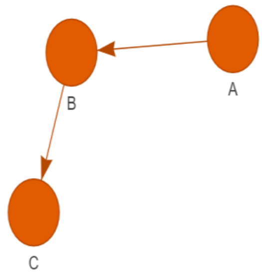
i) **Testons sur un graphe non orienté :**



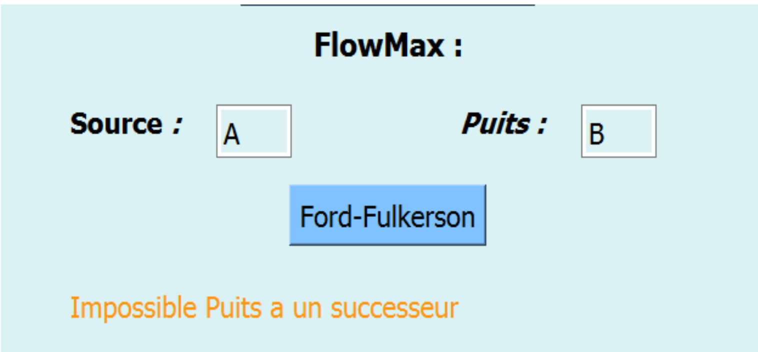



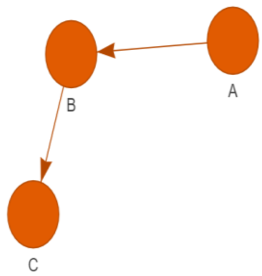

ii) **Testons sur un graphe dont la source S a un prédécesseur:**



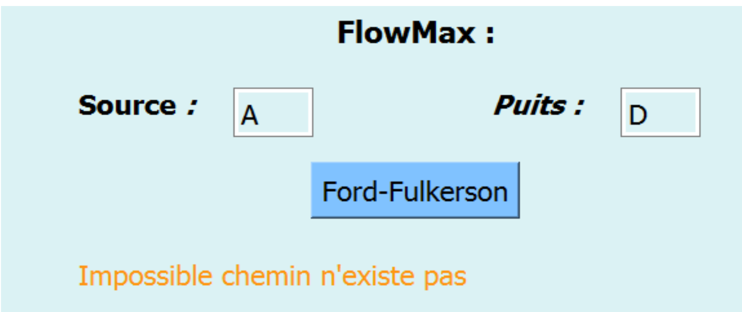



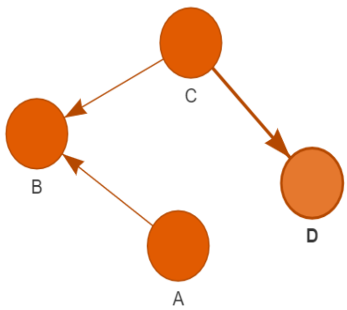

iii) **Testons sur un graphe dont le puits  P a un successeur:**




iv) **Testons sur un graphe où il y a pas de chemin entre la source S et le puits P :**

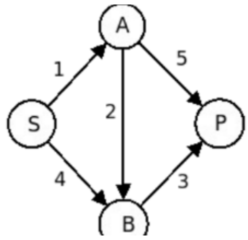


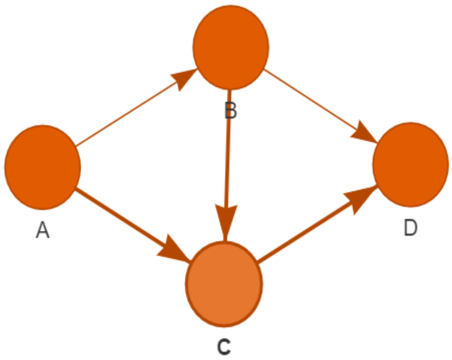



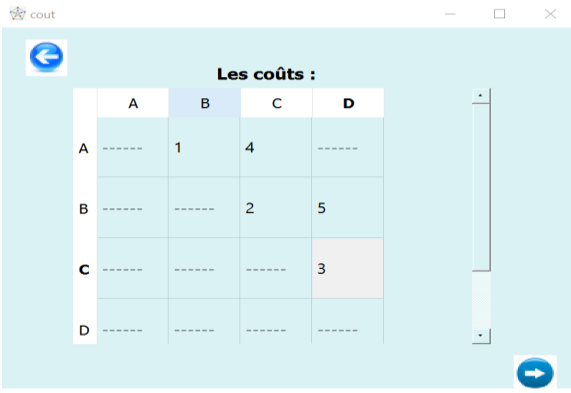
## v) Testons sur un réseau de transport :

- On prend l'exemple du cour :

**Exemple**
Le graphe suivant est bien un réseau de transport car
- le graphe est orienté pondéré sans boucle,
- on a une source S ( pas de prédécesseur de S),
- on a un puits P ( pas de successeur de P),
- on a un chemin de S vers P ( ex. S→A→P)

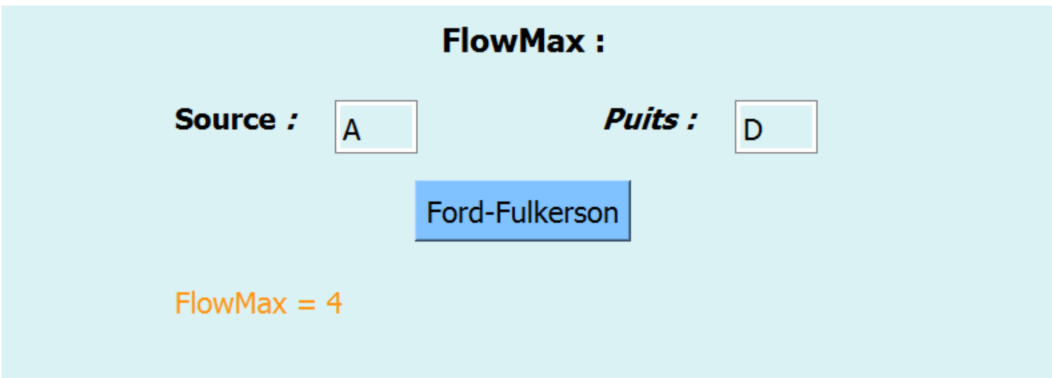



| | A | B | C | D |
|---|---|---|---|---|
| A | ------ | 1 | 4 | ------ |
| B | ------ | ------ | 2 | 5 |
| C | ------ | ------ | ------ | 3 |
| D | ------ | ------ | ------ | ------ |

- **Notre solution :**




# FIN