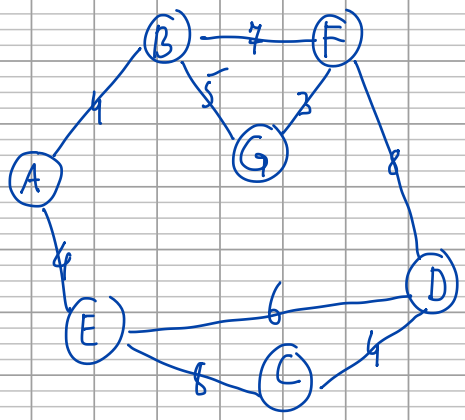
Sujet 0, A 2023

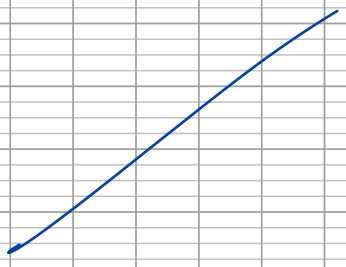
1)



2) A - E - D

3) A pour voisin

	A	B	C	D	E	F	G
A	0	4	0	0	4	0	0
B	4	0	0	0	0	7	5
C	0	0	0	4	8	0	0
D	0	0	4	0	6	8	0
E	4	0	8	6	0	0	0
F	0	7	0	8	0	0	3
G	0	5	0	0	0	3	0

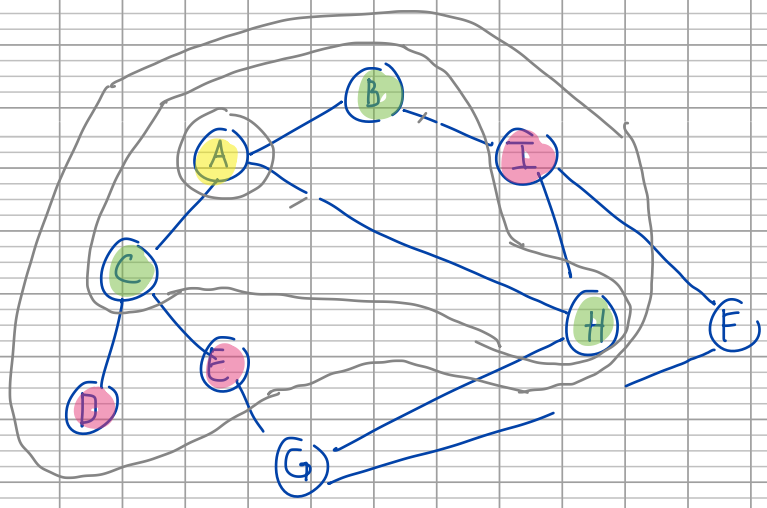


④ g2 = {

```
'A' : ['B', 'C', 'H'],
'B' : ['A', 'I'],
'C' : ['A', 'D', 'E'],
'D' : ['C', 'E'],
'E' : ['C', 'D', 'G'],
'F' : ['G', 'I'],
'G' : ['E', 'F', 'H'],
'H' : ['A', 'G', 'I'],
'I' : ['B', 'F', 'H']
```

}

⑤



A - B - C - H - D - E - I - F - G

⑥ chercher_itineraire est une fonction récursive car elle s'appelle elle même.

⑦ Dans p1, la fonction recherche_itineraires permet de lister tous les iténeraires entre deux sommets. Cette liste est stockée dans la variable tab_itineraires. Il y a donc un effet de bord pour cette fonction puisqu'elle

modifie une variable qui lui est extérieure.

(8) L15 : tab_court = []
L18 : if len(v) <= mini
L19 : mini = v
L22 : tab_court.append(v)

(9) Si on exécute pas à pas le programme p1 entre deux appels de la fonction itinéraire_court, le tableau tab_itinéraires n'est pas réinitialisé comme tableau vide. Ainsi dans le second scénario, tab_itinéraires contient tous les itinéraires
- entre A et E
- entre A et F

Or parmi ces itinéraires, le plus court est ['A', 'C', 'E']. C'est pour cette raison que l'appel itinéraire_court(G2, 'A', 'F') renvoie le résultat énoncé ['A', 'C', 'E'].