

CORRECTION SERIE

GLOWTON

1.

On veut aller de la ville A à la ville D par le chemin le plus court.

Chaque nombre représente la distance entre deux villes.

a.

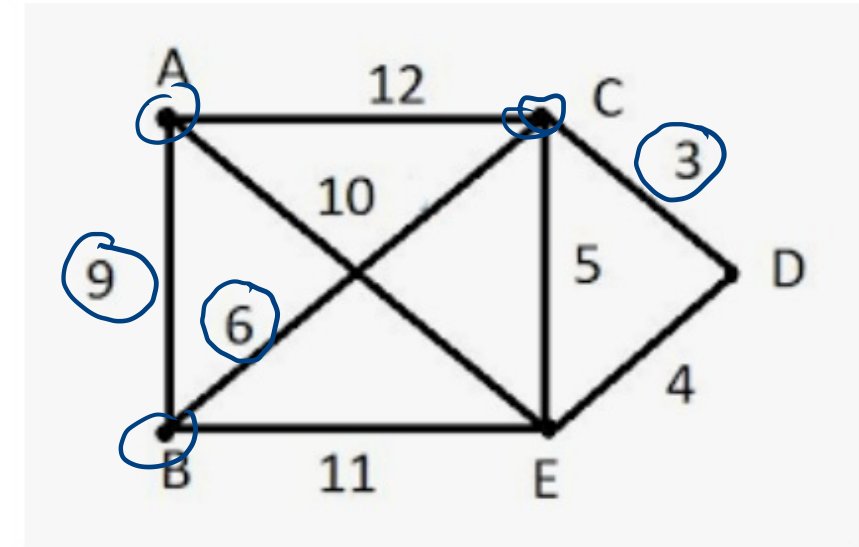
En utilisant un algorithme glouton, quel sera l'itinéraire ?

$A \rightarrow B \rightarrow C \rightarrow D$ distance = 18
↑
glouton: à partir d'une ville, on va vers celle dont la distance est minimale

b.

Y a-t-il une autre solution plus optimale ? Si oui laquelle ?

$A \rightarrow E \rightarrow D$ distance = 14



2. Pour partir en randonnée, on dispose d'un sac à dos. Afin de pouvoir marcher longtemps chaque jour, on décide de limiter la masse totale transportée à 17 kg. Cette limitation ne permet pas de transporter tous les éléments...

On décide de donner une note « utilité » à chacun des objets.

Objet	eau	nourriture	nécessaire de cuisine	couverture	matelas	tente	gaz	lampe	console de jeu	change	jumelle	cartes	pull	machette	livre de chevet
Masse	3	4.3	2.2	1	1.5	3.4	0.8	0.5	2.1	2.8	1.3	0.5	0.8	2.4	0.3
Utilité	20	15	12	6	4	18	13	10	7	11	9	8	8	5	2

On cherche à **maximiser l'utilité totale** des objets emportés dans le sac.

a. Le fait de prendre ou non chaque objet est un choix binaire. En utilisant une approche brutale pour trouver la ou les solutions optimales, combien de cas faudrait-il tester ?

chaque combinaison correspond à un nombre binaire à 15 bits. Ex 100 0000 0000 0001
 on peut faire 2^{15} nombres différents avec 15 bits
 → il y a donc **2^{15} possibilités** (32 768)
 1er objet pris, tous les autres pris, dernier objet pris.

b. Si vous utilisez un algorithme glouton avec comme critère l'optimisation de l'utilité, quelle sera la composition de votre sac à dos ?

glouton ⇒ on maximise l'utilité locale → on essaie de prendre l'objet le plus utile à chaque étape.
 1) eau ($m_t=3$) 4) gaz ($m_t=11,5$)
 2) tente ($m_t=6,4$) 5) nécessaire ($m_t=13,7$)
 3) nourriture ($m_t=10,7$) 6) change ($m_t=16,5$) 7) lampe ($m_t=17$)
 ⚠ Il ne faut pas dépasser 17 kg!
STOP utilité totale 99

c. Y a-t-il d'autres solutions plus optimales ? Si oui donne un exemple.

oui, l'idéal est 117 en prenant

contenu du sac: eau, nécessaire, couverture, tente, gaz, lampe, change, jumelles, cartes, pull, livre

3. On souhaite implémenter un algorithme glouton de rendu de monnaie :

- a. Quelles sont les données qu'il faut prendre en considération ?
- b. Quelle structure de donnée il conviendrait d'utiliser pour chacune ?
- c. Écrivez en pseudo-code un algorithme glouton pour le problème de rendu de monnaie avec $somme = 53$ et $systeme = [0.01, 0.02, 0.05, 0.10, 0.20, 1, 2, 5, 10, 20, 50]$

a) $somme$ (quantité d'argent à rendre)
 $systeme$ (liste mixte des monnaies disponibles)

b) $somme$: float
 $systeme$: liste

c)

C.

Rendu de monnaie (algorithme glouton)

variables

- **entrées:**
 - **système** (liste triée contenant toutes les monnaies existantes)
 - **somme** (nombre correspondant à la somme à rendre)
- **sorties:**
 - **rendu** (liste de toutes les monnaies à rendre)
- autre variable:
 - **monnaie**

pseudo-code

```
ENTREE système
ENTREE somme
rendu = [] (liste vide)
TANT_QUE (somme > 0)
{
    monnaie = système[dernier] (monnaie de plus haute valeur)
    SI (monnaie > somme)
    ALORS
    {
        ENLEVER monnaie DE système
    }
    SINON
    {
        AJOUTER monnaie A rendu
        somme = somme - monnaie
    }
}
SORTIE rendu
```