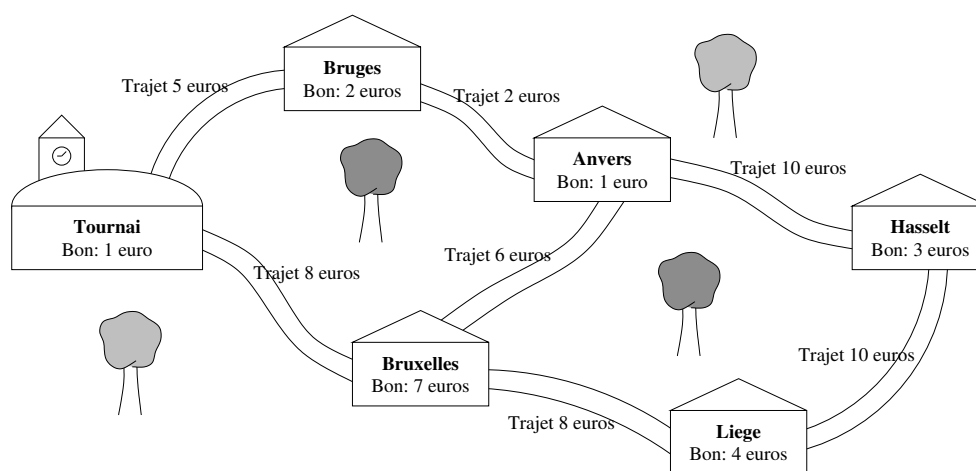


Tâche 1.2 – Trains – (100 pts)

A la suite d'une grève massive de cheminots, la SNCB décide de redorer son image en distribuant des bons de réduction permettant de fortement réduire le coût des trajets. Ces bons sont distribués en gare et, bien que non cumulables, peuvent être utilisés autant de fois que désiré après avoir été collectés. En raison de distensions entre syndicats, régions et autorités communales, ces bons peuvent être différents en chaque gare : la gare d'Anvers peut très bien distribuer des bons de 1 euro alors que la gare de Bruxelles en distribuera de 7 euros.

Les bons collectés peuvent être utilisés sur chaque tronçon de trajet suivant sa collecte, mais sans résulter en une valeur négative. Le bon de 7 euros distribué à Bruxelles pourra donc être utilisé pour le trajet Bruxelles-Leuven à 4 euros, mais ce trajet vous coûtera dès lors 0 euro, on ne vous paiera bien entendu pas 3 euros.

Par exemple, supposons que vous voulez vous rendre de Tournai à Hasselt.



La route la plus directe semble être de passer par Bruges et Anvers. Vous commencez à Tournai où vous collecter un bon de 1 euro, que vous utiliser sur le premier tronçon vers Bruges, qui vous coûte donc $5 - 1 = 4$ euros. A Bruges, vous prenez le bon de 2 euros, et l'utiliser pour voyager vers Anvers gratuitement. Cela ne vous sert à rien de collecter le bon à Anvers vu que vous avez toujours celui de 2 euros que vous pouvez utiliser sur le dernier trajet qui vous coûtera alors $10 - 2 = 8$ euros. Le coût total du trajet sera donc de $4 + 0 + 8 = 12$ euros.

Supposez maintenant que vous passez par Bruxelles pour collecter le bon de 7 euros. Tournai-Bruxelles vous coûtera 7 euros. Vous voyagez ensuite vers Anvers gratuitement et finalement vous rejoignez Hasselt pour 3 euros. Le trajet total vous aura donc coûté 10 euros. Vous pouvez voir que c'est le voyage complet le moins cher que vous pouvez faire.

Tâche

Votre programme doit déterminer quel le coût minimum que peut vous coûter le trajet entre deux villes données.

Limites et contraintes

- $1 \leq n \leq N_{MAX}$, le nombre de gares
- $1 \leq s, f \leq n$, le numéro de station de départ et d'arrivée
- $0 \leq d_i \leq D_{MAX}$, la valeur du bon disponible en gare i
- $1 \leq c_{i,j} \leq C_{MAX}$, le coût du trajet entre la gare i et j , quelque soit le sens et avant que toute réduction lui soit appliquée.

	N_{MAX}	D_{MAX}	C_{MAX}	
Sous-tâche A (32 pts)	50	1 000 000	1 000 000	Tous les bons ont la même valeur.
Sous-tâche B (17 pts)	50	1 000 000	1 000 000	
Sous-tâche C (51 pts)	200	1 000 000	1 000 000	

Durée maximale d'exécution : **0.5 seconde**. Limite mémoire : **128 Mo**.

Entrée

L'entrée donnée à votre programme aura le format suivant :

- La première ligne contient un entier n : le nombre de gare dans le réseau ferroviaire.
- La deuxième ligne contient deux entiers s et f , séparés par un espace. s est la numéro de la station de départ, f est la station à laquelle vous souhaitez arriver.
- La troisième ligne contient n entiers $d_1 d_2 \dots d_n$ séparés par des espaces.
- La quatrième ligne contient l'entier k , le nombre de lignes de chemin de fer.
- Les k lignes suivantes contiennent trois entiers sous la forme $x y c$ qui signifie qu'il y a une ligne entre la gare x et la gare y pour laquelle un trajet coûte c , quelque soit le sens du trajet. Il n'y a pas deux trajets reliant les deux même gares et x est toujours inférieur à y ($1 \leq x < y \leq n$).
- L'entrée se termine par un saut de ligne.

Il est garanti qu'il est toujours possible de rejoindre la gare f à partir de s .

Sortie

La sortie de votre programme contient le coût minimal du trajet entre s et f . Le fichier se termine par un saut de ligne.

Exemple (cfr schéma ci-dessus)

Pour l'entrée suivante

```
6
1 6
1 2 7 1 4 3
7
1 2 5
1 3 8
2 4 2
3 4 6
3 5 8
4 6 10
5 6 10
```

Votre programme renverra :

```
10
```