

INF8775 – Analyse et conception d'algorithmes

TP3 – Hiver 2022

☐ Informations techniques

- ☐ Répondez directement dans le document DOCX.
- ☐ La correction se fait sur ce même rapport.
- ☐ Vous devez faire une remise électronique sur Moodle avant le **20 Avril à 23h59** en suivant les instructions suivantes :
 - Vos fichiers doivent être remis dans une archive zip à la racine de laquelle on retrouve :
 - Le rapport au format DOCX.
 - Un pdf contenant votre analyse asymptotique
 - Un script nommé *tp.sh* servant à exécuter les différents algorithmes du TP. L'interface du script est décrite à la fin du rapport.
 - Le code source et les exécutables.
 - Si le langage que vous utilisez nécessite une phase de compilation, veuillez joindre un Makefile afin que nous puissions le compiler en cas de problème avec vos exécutables. **Si nous ne sommes pas en mesure de tester votre code, vous perdrez des points de respect d'interface et de qualité de code !**
- ☐ Vous avez le choix du langage de programmation utilisé mais vous devrez utiliser les mêmes langage, compilateur et ordinateur pour toutes vos implantations. **Le code et les exécutables soumis devront être compatibles avec les ordinateurs de la salle L-4714.**
- ☐ Si vous utilisez des extraits de codes (programmes) trouvés sur Internet, vous devez en mentionner la source, sinon vous serez sanctionnés pour plagiat.

Mise en situation

Ce dernier travail pratique se fera dans le cadre du concours du meilleur algorithme pour la session d'hiver 2022. Le travail demandé consiste à concevoir et implanter un algorithme **de votre cru** pour résoudre un problème combinatoire. Le classement des équipes déterminera votre note pour la *qualité de l'algorithme*. Votre algorithme sera exécuté sur 3 exemplaires de notre choix pendant 3 minutes chacun.

Ce TP porte sur une simplification d'un problème de physique visant à trouver l'état fondamental d'un cristal.

Soit un graphe $G(V,A)$ où V correspond à un ensemble de sites où seront placés des atomes, et A un ensemble d'arêtes. Le graphe est simple, non pondéré et non orienté. Si l'arête $(i,j) \in F$, alors les sites i et j sont dits voisins. G est un graphe connexe.

Soit N un ensemble d'atomes à placer sur les sites (on suppose que $|N| = |V|$). Il y a k types d'atomes. Notons s_v le type de l'atome sur le site V .

Soit une matrice H , de dimension $k \times k$, indiquant les énergies d'interaction entre voisins. Si deux atomes voisins sont de types i et j , l'énergie associée à ces voisins est $H_{ij} = H_{ji}$. La matrice H est donc symétrique. Les éléments de H peuvent être positifs, négatifs, ou nuls.

Le problème consiste à assigner les atomes de N sur les sites V de manière à minimiser l'énergie totale. L'énergie totale est donnée par :

$$E = \sum_{(i,j) \in F} H_{s_i, s_j}$$

E est donc la somme des énergies d'interaction.

Tous les atomes doivent être placés. Un site ne peut accueillir qu'un seul atome. Un nombre d'atomes de chaque type est donné. Ainsi, si l'instance dit qu'il y a 5 atomes du type 1, alors il faut que la solution contienne 5 atomes de type 1, pas 4 ou 6.

Jeu de données

Nous vous fournissons un générateur d'exemplaires qui fonctionne comme suit :

```
python[3] inst_gen.py -t 1000 -k 3 -n 10
```

Arguments :

- t = nombre de sites et d'atomes (doit être entre 100 et 10000)
- k = nombre de types d'atomes (k doit être entre 2 et 6 compris)
- n = nombre de fichier à créer (optionnel, défaut à 1)

Les instances sont organisées comme suit :

```
<t> <k> <|A|>

<nombre d'atomes de type 1> <nombre d'atomes de type 2> <nombre d'atomes de type 3> .. <nombre d'atomes de type k>

<H11> <H12> <H13> <H14> .. <H1k>
<H21> .. <H2k>
..
..
<Hk1> .. <Hkk>

<noeud a de l'arrête 1> <noeud b de l'arrête 1>
<noeud a de l'arrête 2> <noeud b de l'arrête 2>
..
..
<noeud a de l'arrête |A|> <noeud b de l'arrête |A|>
```

Le fichier d'instances est séparé en 4 sections. Les sections sont séparées par un saut de ligne. Les sections sont :

1. Sur une ligne : le nombre de sites et d'atomes, le nombre de types d'atomes, le nombre d'arêtes du graphe, tous séparés par un espace
2. Sur une ligne : le nombre d'atomes de chaque type, tous séparés par un espace
3. Sur k lignes : la matrice H, les colonnes sont séparées par un espace, les lignes sont séparées par un retour à la ligne
4. Sur |A| lignes : les arêtes. Chaque ligne correspond à une arête du graphe.

Autres critères de correction

Interface d'exécution

Utilisation :

```
tp.sh -e [chemin_vers_exemplaire]
```

Lorsque le script est exécuté sans le paramètre `-p`, le programme affiche uniquement l'énergie E sur une nouvelle ligne, à chaque fois qu'une meilleure solution est trouvée. Votre programme est censé s'exécuter tant et aussi longtemps qu'il n'est pas manuellement interrompu.

Argument optionnel :

`-p` Chaque fois qu'une meilleure solution est trouvée, le programme affiche cette nouvelle solution (**au lieu** d'afficher l'énergie E). Le format de cette solution est décrit ci-dessous

Important : l'option `-e` doit accepter des fichiers avec des chemins absolus.

Format de la solution : Chaque ligne contient une solution. Les solutions sont séparées par un retour à la ligne. Une ligne présente les types des atomes présents dans chaque site, séparés par un espace, dans l'ordre de l'index des sommets. Par exemple, dans la case ci-dessous il y a 3 solutions pour une instance avec 6 types d'atomes et 10 sites . Si nous nous intéressons à la dernière, nous observons que le site 1 possède un atome de type 2, le site 2 possède un atome de type 2, le site 3 possède un atome de type 0, etc...

| |
|---------------------|
| 1 2 4 3 0 1 1 1 2 4 |
| 1 1 3 1 4 1 4 2 2 0 |
| 2 2 0 1 1 4 1 4 1 3 |

Toutes les solutions affichées doivent respecter les contraintes du problème. Mais seul le score de la dernière solution affichée est compté. N'affichez une nouvelle solution que si elle est la meilleure trouvée jusqu'ici !

Un script de vérification de solution vous est fourni (`check_sol.py`, les instructions d'utilisation sont dans le code source). Ce script vous indiquera si l'affichage est correct, si votre solution est valide, ainsi que l'énergie E . C'est ce script qui sera utilisé pour la correction, donc assurez-vous qu'il reconnaisse vos solutions.