

INF4705 — Analyse et conception d’algorithmes

TP 2

Ce travail pratique se répartit sur deux séances de laboratoire et porte sur l’analyse et la conception d’algorithmes développés suivant différents patrons de conception.

1 Problématique

Vous venez de décrocher l’emploi de vos rêves à Vancouver. Malheureusement, le prix des logements dans cette ville est prohibitif et vous auriez du mal à boucler votre budget. Néanmoins, en bon polytechnicien instruit que vous êtes, vous tombez sur l’article suivant :

<http://www.cbc.ca/news/canada/british-columbia/bunker-underground-vancouver-1.4345255>

Voilà, il suffit de creuser un trou et d’y habiter ! Dans un élan d’inspiration, vous vous rappelez qu’un de vos oncles est dynamiteur. En lui empruntant un peu de dynamite, il sera possible de faire un trou sans même utiliser une pelle. De plus, avec la bonne quantité d’explosif, vous pourrez vous rapprocher des conduites d’eau afin d’avoir l’eau courante... Mais attention, si on dépasse cette quantité, on brise les conduites !

Vous avez à votre disposition n bâtons de dynamite de poids respectifs p_1, p_2, \dots, p_n et vous calculez que vous aurez besoin d’un poids total P . Vous devez donc déterminer quelle combinaison de bâtons utiliser pour s’approcher le plus possible du poids total P mais sans le dépasser ! Vous minimiserez ainsi la quantité de terre qu’il reste à pelleter pour atteindre les conduites d’eau.

2 Implantation

Trois algorithmes seront implantés, mettant en pratique les patrons de conception glouton, probabiliste, programmation dynamique et recherche à voisinage variable.

Vous avez le choix du langage de programmation utilisé mais vous devrez utiliser les mêmes langage, compilateur et ordinateur pour toutes vos implantations. N’oubliez pas de spécifier le cadre expérimental (vitesse du CPU, mémoire vive, ...). Le code et les exécutables soumis devront être compatibles avec les ordinateurs de la salle L-4714.

2.1 Algorithme glouton probabiliste

Vous devez concevoir un algorithme glouton de votre cru pour résoudre ce problème. Plutôt qu’être déterministe, il sera probabiliste et devra inclure un aspect aléatoire dans son critère de choix glouton.

2.2 Algorithme de programmation dynamique

Vous devez également résoudre ce problème à l'aide de la programmation dynamique. Soit $f(i, j)$ le poids total le plus élevé qu'on puisse obtenir en combinant certains parmi les i premiers bâtons de dynamite sans dépasser une somme des poids j , qu'on définit récursivement

$$f(i, j) = \max(f(i-1, j-p_i) + p_i, f(i-1, j)).$$

On cherche $f(n, P)$.

2.3 Recherche à voisinage variable

Enfin vous devez implanter l'approche de recherche à voisinage variable avec les trois voisinages suivants :

1. remplacer un bâton de la solution courante par un autre qui n'y est pas ;
2. remplacer un bâton par deux autres ;
3. remplacer deux bâtons par un autre.

Démarrez avec un ensemble de bâtons quelconque (mais valide) et arrêtez quand aucun de vos voisinages n'arrive à améliorer la solution courante.

3 Jeu de données

Les jeux de données du problème se trouvent sur le site du cours Moodle. Les exemplaires sont regroupés par nombre de bâtons n et ordre de grandeur du poids total m : ainsi le fichier "WC-n-m-k.txt" décrit le $k^{\text{ième}}$ exemplaire du groupe.

La structure des fichiers d'exemplaires est :

Sur la première ligne : nombre de bâtons n

Une ligne pour chaque bâton : i <espace> p_i

Sur la dernière ligne : poids total limite P

4 Résultats

Exécutez chacun des trois algorithmes en notant leur temps d'exécution et le poids total, mais ne rapportez dans un tableau que la moyenne de chacune des séries de dix exemplaires. Pour le glouton probabiliste, exécutez chaque exemplaire dix fois et utilisez la moyenne de ces exécutions.

5 Analyse et discussion

1. Tentez une analyse asymptotique du temps de calcul pour chaque algorithme.

2. Servez-vous de vos temps d'exécution pour confirmer et/ou préciser l'analyse asymptotique théorique de vos algorithmes avec la méthode hybride de votre choix (cette méthode peut varier d'un algorithme à l'autre). Justifiez ces choix.
3. Discutez des trois algorithmes en fonction de la qualité respective des solutions obtenues, de la consommation de ressources (temps de calcul, espace mémoire) et de la difficulté d'implantation.
4. Indiquez sous quelles conditions vous utiliseriez l'un de ces algorithmes plutôt que les deux autres.

6 Remise

Avant votre cinquième séance de laboratoire, vous devez faire une *remise électronique* en suivant les instructions suivantes :

1. Le dossier remis doit se nommer `matricule1_matricule2_tp1` et doit être compressé sous format zip.
2. À la racine de ce dernier, on doit retrouver un rapport sous format PDF comprenant :
 - une brève description du sujet et des objectifs de ce travail (svp pas de redite de l'énoncé),
 - la description des jeux de données,
 - la description de votre algorithme glouton,
 - les résultats expérimentaux,
 - l'analyse et discussion
3. Finalement, un script nommé `tp.sh` doit aussi être présent à la racine. Ce dernier sert à exécuter les différents algorithmes du TP. Le fonctionnement du script est décrit à la prochaine section.

7 tp.sh

Utilisation

```
tp.sh -a [glouton | progdyn | vns] -e [path_vers_exemplaire]
```

Arguments optionnels :

- p affiche la solution (rangs des bâtons utilisés) sans texte superflu.
- t affiche le temps d'exécution en ms, sans unité ni texte superflu.

Des exemples de scripts sont disponibles sur Moodle.

8 Barème de correction

- 1 pt** : exposé du travail pratique
- 2 pts** : conception du critère de choix glouton
- 2 pts** : présentation des résultats
- 4 pts** : analyse et discussion
- 2 pts** : les programmes (corrects, structurés, commentés, ...)
- 2 pts** : présentation générale : concision, qualité du français