Algorithmique II

Backtracking – Programmation Dynamique

Exercice 1 : Encore de l'argent.

Nous disposons d'un stock illimité de pièces de monnaie de m valeurs différentes $p_1, p_2, \cdots, p_m$. On veut représenter un montant M avec ces pièces. Par exemple, $M = 12$ pourra être fourni comme $2+3+7$ mais aussi $5+1+6$. Le nombre de représentations de M ne doit donc pas tenir compte de l'ordre des pièces.

1. Soit $P(i, j)$ le nombre de représentations du montant j avec les i premières valeurs du tableau p. Définir une récurrence sur $P(i, j)$.

2. Ecrire alors un algorithme pour calculer $P(u, v)$

3. Analysez votre algorithme.

Exercice 2 : Dominos horizontaux.

Dans cette partie, on considère des dominaux horizontaux avec des lettres donc de la forme $\boxed{\text{a} \mid \text{b}}$. On définit alors une *chaîne* comme une suite dominos telles que les lettres voisines coïncident comme par exemple $\boxed{\text{a} \mid \text{c}}\ \boxed{\text{c} \mid \text{g}}\ \boxed{\text{g} \mid \text{g}}\ \boxed{\text{g} \mid \text{t}}$.

Dans cet exercice, on se donne en **entrée** n dominos (qu'on ne peut pas retourner) $D_1 = \boxed{x_1 \mid y_1}, \cdots, D_n = \boxed{x_n \mid y_n}$.

1. Ecrire un algorithme du type *backtracking* qui à partir des D_i trouve une chaîne si une telle chaîne existe et renvoie une erreur sinon.

2. Ecrire un algorithme du type *programmation dynamique* qui trouve la plus longue sous-séquence de dominos (sans interchanger) qui forme une chaîne.

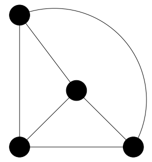
Exercice 3 : MAX-3-COL.

Un graphe est dit 3-colorable si tous les sommets de ce graphe peuvent être colorés en rouge, vert, bleu sans que deux sommets adjacents ne soient voisins.

Dans cette partie, on considère le problème d'optimisation consistant à trouver dans un graphe le plus grand (grand en nombre de sommets et nombre d'arêtes) sous-graphe 3-colorable.

1. En vous inspirant du cours, définir une fonction Score portant sur les sommets et les arêtes d'un graphe pouvant résoudre ce problème d'optimisation.

2. A partir du graphe g donné ci-après dessiner le plus grand sous-graphe sg 3-colorable "extrait" de g. Montrer que votre fonction Score le calcule effectivement.



3. Dans le reste de l'exercice, on ne considère que les graphes de degrés 1, 2 et 3 (chaque sommet a 1 ou 2 ou 3 voisins). Décrire un algorithme qui calcule le score obtenu à partir du plus grand sous-graphe 3-colorable d'un graphe donné en entrée. Montrer sur un petit exemple comment votre algorithme calcule ce score.

4. Quelle est la complexité de votre algorithme ?

Exercice 4 : Contraintes hétérogènes

On a la liste des contraintes suivantes portant sur les entiers naturels A, B, C, D, E et F. $A.C = 1$, $A + B > 0$, $C.D = 0$ $B + C = 0$, $E.F = 0$, $E + D > 0$, $F.D = 1$, $E + B = 0$, $E = D$, $A \leq F$

1. Modéliser le problème sous forme de graphe.

2. En vous inspirant du cours, donner une méthode pour trouver la plus grande liste de contraintes pouvant être satisfaites.