

Impression équilibrée

Informations

Le projet peut être compilé sur Windows (avec Visual studio) et sur Linux (avec la commande *make*), il est réalisé en C++.

1) Sous Windows

Cliquer sur li325.sln pour ouvrir le projet dans Visual studio 2012.

2) Sous Linux

Les sources peuvent être éditées avec n'importe quel éditeur de texte, la compilation s'effectue en utilisant la commande *make* à la racine du projet.

Il est important de comprendre la responsabilité de chaque classe :

1) TList<T>

Il s'agit d'une classe implémentant la structure de liste générique.

2) Range

Cette classe décrit un intervalle, dans notre cas un intervalle de mots.

3) Printer

Cette classe représente l'imprimante.

4) Problem

Il s'agit de la classe qui représente le problème général, elle sert de base pour GloutonProblem et DynamicProblem.

5) GloutonProblem

Une implémentation du problème utilisant un algorithme Glouton.

6) DynamicProblem

Une implémentation du problème utilisant un algorithme Dynamique.

7) String

Une classe facilitant la manipulation des chaînes de caractère.

Les fichiers d'en tête de chacune de ces classes contiennent des commentaires donnant plus de détails sur leur fonctionnement.

Le fichier *main.cpp* créé et résout 12 problèmes, 6 problèmes gloutons et 6 problèmes dynamiques (avec des paramètres différents). Pour cela le programme attend 2 paramètres : le fichier d'entrée et le fichier de sortie. Le fichier d'entrée doit contenir le texte, le fichier de sortie contiendra les solutions ainsi que quelques informations supplémentaires.

Exemple :

li325.exe input.txt output.txt

Questions

- 1) L'algorithme de Glouton est disponible dans la fonction *Solve* de la classe *GloutonProblem*. Cet algorithme ne fournit pas l'optimum, voici un contre-exemple :

Algorithme de Glouton appliqué au texte avec $W = 20$

LINES	BADNESS

Inventée_par_le_____	125
professeur_Richard__	8
Bellman,_la_____	729
programmation_____	343
dynamique_permet_de_	1
résoudre_au_moyen____	27
d'un_ordinateur_tout	0
problème_____	1728
.....	

Le score total peut être amélioré:

de

.....	
d'un_ordinateur_tout	0
problème_____	1728
.....	

à

.....	
d'un_ordinateur_____	125
tout_problème_____	343
.....	

L'algorithme de glouton nous fournit donc une heuristique.

- 2) L'algorithme Dynamique est disponible dans la fonction *Solve* de la classe *DynamicProblem*.

Notations :

W : capacité d'une ligne.

$Words$: tableau de mots.

n : taille du tableau $Words$.

$Badness$: fonction du calcul du score, plus il est bas, meilleur il est.

$DP[i]$: structure optimale, contient le score *Badness* minimum réalisable en prenant en compte les mots $Words[i : n]$.

- a) Analyse de la complexité temporelle :

Nous cherchons donc à calculer $DP[0]$, le tableau DP est défini comme tel :

$$\begin{cases} DP[n+1] = 0 \\ DP[i] = \min_{j \in [i+1, W]} (Badness(i, j) + DP[j+1]) \end{cases}$$

Le calcul du *min* se fait en $O(W)$.

Le remplissage du tableau *DP* se fera en $O(n)$ maximum (Si la valeur demandée y est absente, alors un appel récursif est engendré).

Cela nous donne un total de $O(Wn)$.

Remarque : La complexité est en fait inférieure puisque le calcul du *min* est restreint au nombre maximum de mots entrants sur une ligne.

b) Analyse de la complexité en mémoire :

La structure optimale ne demande qu'un tableau de n entiers.

Au plus n appels récursifs seront exécutés.

Au total ($n*4$ + taille de la fonction *DP*) octets seront nécessaires.

3) Ne prendre que la somme des caractères améliorera le temps nécessaire au calcul du score *Badness*. Cela ne change rien pour l'algorithme résolvant le problème.

4) L'algorithme n'a pas eu besoin de modification, seuls les paramètres du problème ont changés (voir annexe).

Annexe

Voici les différents résultats retournés pour le texte et les paramètres demandés :

Glouton Problem W=20 Power=1

LINES BADNESS

Inventée_par_le_____ 5

professeur_Richard__ 2

Bellman,_la_____ 9

programmation_____ 7

dynamique_permet_de_ 1

résoudre_au_moyen___ 3

d'un_ordinateur_tout 0

problème_____ 12

d'optimisation_dont_ 1

la_fonction_objectif 0

se_décrit_comme_la__ 2

somme_de_fonctions__ 2

monotones_____ 11

non-décroissantes__	3
des_ressources.____	5
Concrètement,_cela__	2
signifie_que_l'on_va	0
pouvoir_déduire_la__	2
solution_optimale__	3
d'un_problème_à_____	5
partir_d'une_____	8
solution_optimale__	3
d'un_sous_problème._	1
On_appelle_____	10
algorithme_glouton__	2
un_algorithme_qui__	3
suit_le_principe_de_	1
faire,_étape_par_____	4
étape,_un_choix_____	5
optimum_local,_dans_	1
l'espoir_d'obtenir__	2
un_résultat_optimum_	1
global._Dans_les_cas	0
où_l'algorithme_ne__	2
fournit_pas_____	9
systématiquement_la_	1
solution_optimale,___	2
il_est_appelé_une__	3
heuristique_____	9
gloutonne._____	0

TOTAL	142

Glouton Problem W=20 Power=2

LINES BADNESS

Inventée_par_le_____	25
professeur_Richard__	4
Bellman,_la_____	81
programmation_____	49

dynamique_permet_de_	1
résoudre_au_moyen__	9
d'un_ordinateur_tout	0
problème_____	144
d'optimisation_dont_	1
la_fonction_objectif	0
se_décrit_comme_la__	4
somme_de_fonctions__	4
monotones_____	121
non-décroissantes__	9
des_ressources.____	25
Concrètement,_cela__	4
signifie_que_l'on_va	0
pouvoir_déduire_la__	4
solution_optimale__	9
d'un_problème_à_____	25
partir_d'une_____	64
solution_optimale__	9
d'un_sous_problème._	1
On_appelle_____	100
algorithme_glouton__	4
un_algorithme_qui____	9
suit_le_principe_de_	1
faire,_étape_par_____	16
étape,_un_choix_____	25
optimum_local,_dans_	1
l'espoir_d'obtenir__	4
un_résultat_optimum_	1
global._Dans_les_cas	0
où_l'algorithme_ne__	4
fournit_pas_____	81
systématiquement_la_	1
solution_optimale,___	4
il_est_appelé_une____	9
heuristique_____	81
gloutonne._____	0

Glouton Problem W=20 Power=3

LINES BADNESS

Inventée_par_le_____	125
professeur_Richard__	8
Bellman,_la_____	729
programmation_____	343
dynamique_permet_de_	1
résoudre_au_moyen__	27
d'un_ordinateur_tout	0
problème_____	1728
d'optimisation_dont_	1
la_fonction_objectif	0
se_décrit_comme_la__	8
somme_de_fonctions__	8
monotones_____	1331
non-décroissantes__	27
des_ressources._____	125
Concrètement,_cela__	8
signifie_que_l'on_va	0
pouvoir_déduire_la__	8
solution_optimale__	27
d'un_problème_à_____	125
partir_d'une_____	512
solution_optimale__	27
d'un_sous_problème._	1
On_appelle_____	1000
algorithme_glouton__	8
un_algorithme_qui__	27
suit_le_principe_de_	1
faire,_étape_par_____	64
étape,_un_choix_____	125
optimum_local,_dans_	1
l'espoir_d'obtenir__	8
un_résultat_optimum_	1
global._Dans_les_cas	0

où_l'algorithme_ne__	8
fournit_pas_____	729
systématiquement_la_	1
solution_optimale,___	8
il_est_appelé_une___	27
heuristique_____	729
gloutonne._____	0

TOTAL	7906

Dynamic Problem W=20 Power=1

LINES	BADNESS

Inventée_____	12
par_le_professeur__	3
Richard_Bellman,___	4
la_programmation___	4
dynamique_permet___	4
de_résoudre_au_moyen	0
d'un_ordinateur_____	5
tout_problème_____	7
d'optimisation_____	6
dont_la_fonction_____	4
objectif_se_décrit__	2
comme_la_somme_de___	3
fonctions_monotones_	1
non-décroissantes___	3
des_ressources._____	5
Concrètement,_____	7
cela_signifie_____	7
que_l'on_va_pouvoir_	1
déduire_la_solution_	1
optimale_d'un_____	7
problème_à_partir___	3
d'une_solution_____	6
optimale_d'un_sous___	2
problème._On_appelle	0

algorithmes_glouton__	2
un_algorithme_____	7
qui_suit_le_principe	0
de_faire,_étape_____	5
par_étape,_un_choix_	1
optimum_local,_dans_	1
l'espoir_d'obtenir__	2
un_résultat_optimum_	1
global._Dans_les_____	4
cas_où_l'algorithme_	1
ne_fournit_pas_____	6
systématiquement_la_	1
solution_optimale,___	2
il_est_appelé_____	7
une_heuristique_____	5
gloutonne._____	0

TOTAL	142

Dynamic Problem W=20 Power=2

LINES	BADNESS

Inventée_par_____	64
le_professeur_____	49
Richard_Bellman,___	16
la_programmation___	16
dynamique_permet_de_	1
résoudre_au_moyen___	9
d'un_ordinateur_____	25
tout_problème_____	49
d'optimisation_____	36
dont_la_fonction_____	16
objectif_se_décrit__	4
comme_la_somme_de___	9
fonctions_monotones_	1
non-décroissantes___	9
des_ressources._____	25

Concrètement, _____	49
cela_signifie_que____	9
l'on_va_pouvoir_____	25
déduire_la_solution_	1
optimale_d'un_____	49
problème_à_partir____	9
d'une_solution_____	36
optimale_d'un_sous__	4
problème._On_appelle	0
algorithme_glouton__	4
un_algorithme_qui____	9
suit_le_principe_____	16
de_faire,_étape_____	25
par_étape,_un_choix_	1
optimum_local,_dans_	1
l'espoir_d'obtenir____	4
un_résultat_optimum_	1
global._Dans_les_____	16
cas_où_l'algorithme_	1
ne_fournit_pas_____	36
systématiquement_la_	1
solution_optimale,____	4
il_est_appelé_____	49
une_heuristique_____	25
gloutonne._____	0

TOTAL	704

Dynamic Problem W=20 Power=3

LINES	BADNESS

Inventée_par_____	512
le_professeur_____	343
Richard_Bellman,____	64
la_programmation____	64
dynamique_permet_de_	1
résoudre_au_moyen____	27

d'un_ordinateur_____	125
tout_problème_____	343
d'optimisation_____	216
dont_la_fonction_____	64
objectif_se_décrit__	8
comme_la_somme_de___	27
fonctions_monotones_	1
non-décroissantes___	27
des_ressources._____	125
Concrètement,_____	343
cela_signifie_que___	27
l'on_va_pouvoir_____	125
déduire_la_solution_	1
optimale_d'un_____	343
problème_à_partir___	27
d'une_solution_____	216
optimale_d'un_sous___	8
problème._On_appelle	0
algorithme_glouton__	8
un_algorithme_qui___	27
suit_le_principe_____	64
de_faire,_étape_____	125
par_étape,_un_choix_	1
optimum_local,_dans_	1
l'espoir_d'obtenir__	8
un_résultat_optimum_	1
global._Dans_les_____	64
cas_où_l'algorithme_	1
ne_fournit_pas_____	216
systématiquement_la_	1
solution_optimale,___	8
il_est_appelé_____	343
une_heuristique_____	125
gloutonne._____	0

TOTAL	4030

LINES	BADNESS

Inventée par le professeur Richard Bellman, la	4
programmation dynamique permet de résoudre au	5
moyen d'un ordinateur tout problème d'optimisation	0
dont la fonction objectif se décrit comme la somme	0
de fonctions monotones non-décroissantes des	6
ressources. Concrètement, cela signifie que l'on	2
va pouvoir déduire la solution optimale d'un	6
problème à partir d'une solution optimale d'un	4
sous problème. On appelle algorithme glouton un	3
algorithme qui suit le principe de faire, étape	3
par étape, un choix optimum local, dans l'espoir	2
d'obtenir un résultat optimum global. Dans les cas	0
où l'algorithme ne fournit pas systématiquement la	0
solution optimale, il est appelé une heuristique	2
gloutonne. _____	0

TOTAL	37

Glouton Problem W=50 Power=2

LINES	BADNESS

Inventée par le professeur Richard Bellman, la	16
programmation dynamique permet de résoudre au	25
moyen d'un ordinateur tout problème d'optimisation	0
dont la fonction objectif se décrit comme la somme	0
de fonctions monotones non-décroissantes des	36
ressources. Concrètement, cela signifie que l'on	4
va pouvoir déduire la solution optimale d'un	36
problème à partir d'une solution optimale d'un	16
sous problème. On appelle algorithme glouton un	9
algorithme qui suit le principe de faire, étape	9
par étape, un choix optimum local, dans l'espoir	4
d'obtenir un résultat optimum global. Dans les cas	0
où l'algorithme ne fournit pas systématiquement la	0

solution_optimale,_il_est_appelé_une_heuristique__	4
gloutonne._____	0

TOTAL	159

Glouton Problem W=50 Power=3

LINES	BADNESS

Inventée_par_le_professeur_Richard_Bellman,_la__	64
programmation_dynamique_permet_de_résoudre_au__	125
moyen_d'un_ordinateur_tout_problème_d'optimisation	0
dont_la_fonction_objectif_se décrit_comme_la_somme	0
de_fonctions_monotones_non-décroissantes_des__	216
ressources._Concrètement,_cela_signifie_que_l'on__	8
va_pouvoir_déduire_la_solution_optimale_d'un__	216
problème_à_partir_d'une_solution_optimale_d'un__	64
sous_problème._On_appelle_algorithme_glouton_un__	27
algorithme_qui_suit_le_principe_de_faire,_étape__	27
par_étape,_un_choix_optimum_local,_dans_l'espoir__	8
d'obtenir_un_résultat_optimum_global._Dans_les_cas	0
où_l'algorithme_ne_fournit_pas_systématiquement_la	0
solution_optimale,_il_est_appelé_une_heuristique__	8
gloutonne._____	0

TOTAL	763

Dynamic Problem W=50 Power=1

LINES	BADNESS

Inventée_par_le_professeur_Richard_Bellman,_____	7
la_programmation_dynamique_permet_de_résoudre_au__	2
moyen_d'un_ordinateur_tout_problème_d'optimisation	0
dont_la_fonction_objectif_se décrit_comme_____	9
la_somme_de_fonctions_monotones_non-décroissantes_	1
des_ressources._Concrètement,_cela_signifie_____	7
que_l'on_va_pouvoir_déduire_la_solution_optimale__	2

d'un problème à partir d'une solution optimale____	4
d'un sous problème. On appelle algorithme glouton	1
un algorithme qui suit le principe de faire, étape	0
par étape, un choix optimum local, dans l'espoir__	2
d'obtenir un résultat optimum global. Dans les cas	0
où l'algorithme ne fournit pas systématiquement la	0
solution optimale, il est appelé une heuristique__	2
gloutonne. _____	0

TOTAL	37

Dynamic Problem W=50 Power=2

LINES	BADNESS

Inventée par le professeur Richard Bellman, la____	16
programmation dynamique permet de résoudre au____	25
moyen d'un ordinateur tout problème d'optimisation	0
dont la fonction objectif se décrit comme la____	36
somme de fonctions monotones non-décroissantes____	16
des ressources. Concrètement, cela signifie que__	9
l'on va pouvoir déduire la solution optimale d'un	1
problème à partir d'une solution optimale d'un____	16
sous problème. On appelle algorithme glouton un__	9
algorithme qui suit le principe de faire, étape____	9
par étape, un choix optimum local, dans l'espoir__	4
d'obtenir un résultat optimum global. Dans les cas	0
où l'algorithme ne fournit pas systématiquement la	0
solution optimale, il est appelé une heuristique__	4
gloutonne. _____	0

TOTAL	145

Dynamic Problem W=50 Power=3

LINES	BADNESS

Inventée par le professeur Richard Bellman, la____	64

programmation_dynamique_permet_de_résoudre_au_____	125
moyen_d'un_ordinateur_tout_problème_d'optimisation	0
dont_la_fonction_objectif_se_décrit_comme_la_____	216
somme_de_fonctions_monotones_non-décroissantes_____	64
des_ressources._Concrètement,_cela_signifie_que_____	27
l'on_va_pouvoir_déduire_la_solution_optimale_d'un_	1
problème_à_partir_d'une_solution_optimale_d'un_____	64
sous_problème._On_appelle_algorithme_glouton_un_____	27
algorithme_qui_suit_le_principe_de_faire,_étape_____	27
par_étape,_un_choix_optimum_local,_dans_l'espoir_____	8
d'obtenir_un_résultat_optimum_global._Dans_les_cas	0
où_l'algorithme_ne_fournit_pas_systématiquement_la	0
solution_optimale,_il_est_appelé_une_heuristique_____	8
gloutonne._____	0

TOTAL	631

On remarque que l'algorithme Glouton est insensible à la valeur du paramètre *power* (évidant).

L'algorithme dynamique, qui nous retourne l'optimum est sensible au paramètre *power*, plus celui-ci est élevé moins violents sont les espacements (ils sont plus réguliers) ce qui est intéressant si l'on souhaite « étendre » chaque ligne en espaçant plus ou moins les mots.

Je me suis inspiré du cours suivant : <http://www.youtube.com/watch?v=ENyox7kNKeY>.