Enseignement en Programmation Sportive
EPS

TD3 - Programmation dynamique

Joël Felderhoff, Quoc Tung Le, Lucas Perotin, Eric Thierry

Le menu du jour

Séance $N = 3$:

- Debriefing $N - 3$: prochaine fois (après avoir vu vos codes)
- Solution $N - 2$: TD1 gestion I/O, premières SD
- Discussion $N - 1$: TD2 énumération, glouton, dicho
- Présentation N : TD3 prog dyn (= smart recursion)

L'algo du jour : Longest Increasing Sequence (LIS)

Solution : TD1 gestion I/O, premières SD

Problème OJ 10055 : Hashmat the Brave Warrior

Calcul de la distance entre deux nombres

Solution : gestion I/O

Entrainement à la gestion I/O, en particulier les entrées/sorties streamées utilisées par les serveurs Juges. Petits pièges :

- la taille des entiers à traiter : pas de souci en Python, utiliser `long long` en C++ pour 64 bits garantis (`int` pas suffisant avec 32 bits, `long` risqué car 32 ou 64 bits suivant OS)
- les deux entiers n'étaient pas toujours triés (contrairement à l'exemple) donc faire un calcul de valeur absolue de la différence.

Solution : TD1 gestion I/O, premières SD

Problème OJ 272 : TEX Quotes

Text processing : remplacement de ponctuation (quotation marks).

Solution : gestion I/O

Entrainement à la gestion I/O, en particulier les entrées/sorties streamées utilisées par les serveurs Juges. Possibilité d'utiliser les options offertes avec `cin` comme `noskipws` pour la lecture caractère par caractère (cf "cours" sur site web).

Solution : TD1 gestion I/O, premières SD

Problème OJ 11559 : Event Planning

Calcul glouton de minimum sur des données filtrées par des contraintes simples (inégalités entre entiers)

Solution

Entrainement à la gestion I/O avec un autre format d'entrée pouvant être lu avec `cin` en C++ et trois boucles imbriquées.

Solution : TD1 gestion I/O, premières SD

Problème OJ 10420 : List of Conquests

Comptage d'objets, ici des strings (les noms des pays)

Solution

Entrainement à la gestion I/O avec encore un autre format d'entrée et exemple d'utilisation d'un dictionnaire comme structure de données (SD) : `dict` en Python, `map<string,int>` en C++ dans notre cas ici.

Solution : TD1 gestion I/O, premières SD

Problème OJ 12205 : Happy Telephones

Comptage d'intersections entre intervalles.

Solution : glouton

Entrainement à la gestion I/O avec un autre format d'entrée et utilisation de tableau, p.ex. tableau dynamique `vector<int>` pour stocker juste un entier ou `vector<pair<int,int>>` pour stocker une paire d'entiers (les extremités de chaque intervalle).

Discussion : échelle de temps

Instructions Per Second : une mesure de performance
(1 GIPS = 10^9 instructions/second)

Exemples d'ordre de grandeur : temps d'exécution, sur une machine à 300 GIPS, d'un programme dont le nombre d'opérations élémentaires est estimé à $op(n)$ pour une entrée de taille n.

complexity	**data size n**					
$\downarrow op(n) \downarrow$	$n = 10$	$n = 20$	$n = 50$	$n = 100$	$n = 10^6$	$n = 10^{12}$
$\log_2(n)$	1.10^{-11} s	1.10^{-11} s	2.10^{-11} s	2.10^{-11} s	6.10^{-11} s	1.10^{-10} s
n	3.10^{-11} s	7.10^{-11} s	1.10^{-10} s	3.10^{-8} s	3.10^{-6} s	3 s
n^2	3.10^{-10} s	1.10^{-9} s	8.10^{-9} s	3.10^{-8} s	3 s	10^5 y
2^n	3.10^{-9} s	4.10^{-6} s	1 h	10^{11} y	$> 10^{100}$ y	$> 10^{100}$ y
$n!$	1.10^{-5} s	94 d	3.10^{45} y	$> 10^{100}$ y	$> 10^{100}$ y	$> 10^{100}$ y

s = second, m = minute, h = hour, d = day, y = year

Discussion : échelle de temps

La complexité algorithmique en pratique :

- Vrai temps d'exécution à mieux analyser (opérations pas si élémentaires, parallélisme à tous les niveaux, hiérarchie des mémoires, environnement d'exécution niveau OS ...), mais ordres de grandeur utiles à connaître en première estimation
- Nuancer complexité pire vs complexité en moyenne, des algos non optimaux en complexité pire cas asymptotique peuvent être plus performants sur certains cas pratiques, néanmoins pire cas possibles car les concepteurs d'exos en prog sportive sont parfois un peu vicieux
- Analyse théorique (au moins à la louche) = bonne aide à la décision pour choisir entre plusieurs premières approches