

TD 1 Complexité

Luc Lapointe

`luc.lapointe@ens-paris-saclay.fr`

`home.lmf.cnrs.fr/LucLapointe/`

On rappelle quelques définitions de classes de complexité usuelles :

- $L = \text{SPACE}(\log n)$ est la classe des langages décidés par une machine de Turing déterministe utilisant un espace logarithmique.
- $P = \bigcup_{k \in \mathbb{N}} \text{TIME}(n^k) = \text{TIME}(n^{O(1)})$ est la classe des langages décidés par une machine de Turing déterministe en temps polynomial.

On rappelle que l'espace utilisé par une machine de Turing est celui utilisé par sa bande de travail.

Exercice 1 - Comparaison de deux entiers

Trouver des algorithmes les plus simples possibles, en terme de clarté, parmi les algorithmes en temps polynomial qui répondent aux questions suivantes. Préciser leur complexité en temps et en espace.

1. **Entrée** : deux entiers n et k représentés en base 2 séparés par $\#$.

Question : est-ce que $n = k$?

2. **Entrée** : deux entiers n et k représentés en base 2 séparés par $\#$.

Question : est-ce que $n \leq k$?

Exercice 2 - Addition

Trouver un algorithme en espace logarithmique répondant à la question suivante :

Entrée : deux entiers n et k représentés en base 2 séparés par $\#$.

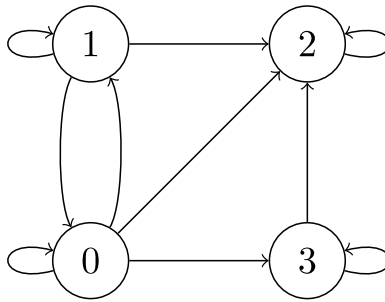
Sortie : $n + k$?

Exercice 3 - Comparaison en espace logarithmique

Trouver des algorithmes en espace logarithmique pour les problèmes de l'exercice 1.

Exercice 4 - Représentation des graphes

Un graphe orienté est un couple $G = (S, A)$ où $S = \{0, \dots, n-1\}$ est un ensemble fini de n sommets et $A \subseteq S \times S$ un ensemble d'arêtes. Il existe deux façons standards de représenter un graphe avec un alphabet fini Σ contenant au moins $\{0, 1\}$: les *listes d'adjacence* et les *matrices d'adjacence*.



1. Décrire ce que sont ces représentations.
2. Proposer un alphabet Σ utilisable pour ces deux représentations de graphes.
3. Décrire le graphe ci-dessus en liste et en matrice d'adjacence avec cet alphabet.
4. Montrer qu'il existe une fonction logspace (i.e. qui peut être implémentée par une MT qui utilise un espace logarithmique) qui à toute représentation d'un graphe G en liste d'adjacence associe une représentation de G en matrice d'adjacence.
Réciproquement, montrer qu'il existe une fonction logspace qui à toute représentation d'un graphe G en matrice d'adjacence associe une représentation de G en liste d'adjacence.

Exercice 5 - Temps polynomial exact

On appelle *monôme* une fonction de la forme $x \mapsto \lambda x^k$ où $\lambda, k \in \mathbb{N}$. On rappelle qu'une fonction f est dite *constructible* s'il existe une machine M qui produit, pour toute entrée de longueur n , la représentation unaire de $f(n)$ en temps $O(n + f(n))$ et en espace $O(f(n))$.

1. Montrer que tout monôme est constructible.
2. Montrer que pour f constructible et telle que $f(n) \geq n$, il existe $L \in \mathbb{N}$ tel que f est constructible en temps *exactement* $Lf(n)$.
3. Définissons POLY' comme la classe des langages L tels qu'il existe une MT M et un polynôme p avec M décidant L en temps exactement p (au lieu de au plus p). Montrer que $\text{POLY}' = \text{P}$.

Exercice 6 - Temps d'arrêt de L

1. Montrer que toute MT qui calcule en espace logarithmique s'arrête en temps polynomial.
2. Définissons L' comme la classe des langages semi-décidés en espace logarithmique (i.e. la machine peut ne pas s'arrêter sur x si x n'est pas dans le langage). Montrer que $L' = L$.

Exercice 7 - Composition de fonctions logspace

Soient $h_1 : L_1 \mapsto L_2$ et $h_2 : L_2 \mapsto L_3$ deux fonctions calculables en espace logarithmique par des machines déterministes.

1. Donner la machine naïve calculant $h_2 \circ h_1$. Quel est l'espace utilisé par cette machine ?
2. Montrer que la fonction $h_2 \circ h_1 : L_1 \mapsto L_3$ peut aussi être calculée en espace logarithmique par une machine déterministe.
3. Rappeler un intérêt majeur de cette propriété.