

IPcute : machine virtuelle et analyse de la complexité d'algorithmes

François BRET
Département d'Informatique
UFR Sciences et Techniques - Université de Tours
Parc Grandmont 37200 TOURS France
bret@univ-tours.fr

Résumé

L'apprentissage de l'algorithmique est une activité difficile pour l'étudiant. L'étude de la complexité d'un algorithme ou d'une famille d'algorithmes, qui constitue un point important de cette discipline, est souvent mal comprise par les étudiants, en particulier lorsqu'elle aboutit à la résolution d'équations mathématiques récurrentes complexes. On propose ici de faciliter cet apprentissage par l'utilisation du logiciel IPcute, qui permet l'étude visuelle et interactive d'algorithmes créés par l'étudiant. En choisissant différents jeux de données, l'étudiant peut comparer graphiquement les courbes de complexité d'une famille d'algorithmes. De plus, IPcute détermine la complexité asymptotique moyenne des algorithmes étudiés d'après les jeux de données fournis.

Abstract

Learning algorithmics is difficult for students. The study of the complexity of an algorithm or a group of algorithms is an important aspect of this course, but not always well understood by the student, particularly when complicated recurrent equations have to be resolved. In this paper, we propose to facilitate the learning process by using the IPcute software. The algorithms, prepared by the student, are represented visually and interactively. By selecting different data sets, the student can compare graphically the complexity curves of several algorithms. Moreover, IPcute gives the mean asymptotic complexity of these algorithms, depending on the data sets provided.

Mots clés : algorithmique, didactique de l'informatique, EAO, NTE, pédagogie à l'université, machine virtuelle, complexité.

Keywords : algorithmics, didactic of informatics, CAL, NTEs, pedagogy in higher education, virtual machine, complexity.

1. Introduction

L'étude de la complexité des algorithmes fait partie intégrante de tout cours d'algorithmique et a pour but d'évaluer le temps d'exécution et/ou l'espace alloué par un algorithme donné. De plus, elle permet leurs comparaisons et leurs rattachements à une famille de complexité. Elle donne souvent lieu à l'établissement d'équations récurrentes dans le cas d'algorithmes récursifs, qui nécessitent parfois des calculs non triviaux pour

parvenir à leur résolution ou à un encadrement de leurs solutions. Les étudiants établissent difficilement ces équations, et, quand il les obtiennent, n'arrivent pas à les résoudre correctement, ni même à avoir une idée sur le type de croissance des fonctions solutions.

On propose de montrer ici comment il est possible de rendre plus accessible l'analyse des algorithmes par une étude expérimentale en utilisant le système pédagogique de développement IPcute (décrit dans les actes de TICE2000) et de la machine virtuelle associée. Le système IPcute permet actuellement l'écriture et l'exécution interprétée de programmes en Pascal de manière visuelle et interactive avec l'utilisateur.

2. Présentation de la machine virtuelle

IPcute permet de compiler les sources des programmes présents dans l'éditeur en instructions de langage machine d'une machine virtuelle. Cette dernière comprend actuellement 11 registres accessibles d'adresses de 32 bits chacun, permettant d'adresser la mémoire externe en quantité suffisante. Les piles d'exécution et d'évaluation sont des zones de mémoire externe ou interne selon l'implémentation choisie pour le processeur.

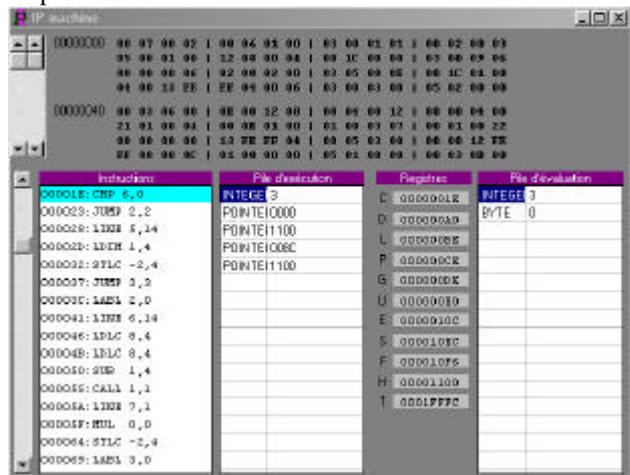


Fig. 1 La machine virtuelle

Un compteur d'instructions exécutées est présent dans la machine virtuelle, ce qui permet d'avoir une mesure de temps écoulé depuis le début d'exécution du

programme, indépendante de la fréquence réelle du processeur virtuel.

3. Analyse de la complexité des algorithmes

Une partie de l'algorithmique consiste à étudier et à comparer ces algorithmes dans leurs dimensions mesurables, principalement le temps d'exécution, parfois l'espace mémoire nécessaire. Ces mesures sont généralement dépendantes de la taille des données initiales dans le nombre de données fournies, voire dans la magnitude de ces données. Il s'agit donc de déterminer une fonction définie sur le domaine des mesures choisies représentant les données initiales.

Au lieu d'étudier la source de l'algorithme pour en déduire des équations de complexité, on peut aussi implémenter l'algorithme sur une machine et mesurer le temps d'exécution pour différentes données initiales parcourant l'ensemble du domaine des mesures qu'on y a associées. Ainsi il est possible, en analysant la suite des points (mesure, temps d'exécution) de retrouver l'allure de la courbe de complexité correspondant à l'algorithme étudié. Ceci permet d'obtenir une indication sur le résultat obtenu par le calcul mathématique de la complexité.

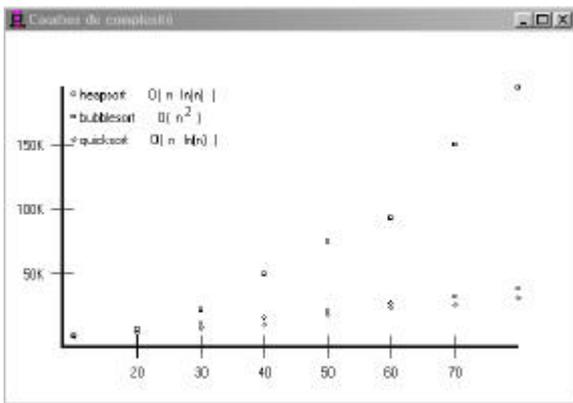


Fig. 2a Complexité d'algorithmes de tris sur des données initiales au hasard

IPcute permet le tracé de plusieurs courbes de complexité sur le même graphique, donnant une indication visuelle immédiate du comportement d'algorithmes distincts traitant un même jeu de données. Chaque courbe est représentée point par point par une couleur distincte et un symbole approprié, accompagnée d'une légende choisie par l'utilisateur et de la fonction complexité temps estimée à partir de l'ensemble des points de la courbe. Notons que ce résultat reste indicatif et ne remplace pas un calcul mathématique de complexité qui n'est pas implémenté dans IPcute.

L'utilisateur, désirant obtenir la visualisation d'une ou plusieurs courbes de complexité, doit seulement placer,

au début du corps des procédures ou fonctions à visualiser, l'appel à la procédure :

complexity(n , message)

avec n expression entière correspondant à la mesure du jeu de données, valeur généralement calculée en considérant un ou plusieurs paramètres de la procédure ou fonction à visualiser, et message une chaîne de caractères (souvent le nom de la procédure) correspondant à la légende de la courbe dans le graphique. Il n'est pas possible de considérer plusieurs mesures simultanément pour un même algorithme dans IPcute.

Le temps d'exécution d'un algorithme sous IPcute ne dépend pas d'une horloge système, mais du nombre d'instructions machine exécutées dans la machine virtuelle. Ainsi, même si l'exécution a été arrêtée par un point d'arrêt ou par un autre processus machine plus prioritaire, la courbe de complexité reste significative.

4. Un exemple pédagogique: la comparaison d'algorithmes de tris

On s'intéresse aux algorithmes de tris suivants : tri à bulles (bubblesort) , tri rapide (quicksort), tri en tas (heapsort). Qu'on applique à un tableau de N nombres entiers, choisis au hasard, (N variant de 10 à 80 par pas de 10), et on applique ce tableau à chacun des algorithmes de tris, dans chacun desquels on a ajouté l'instruction complexity, dans laquelle le paramètre associé à la mesure est N et le message associé est le nom du tri. On obtient alors le graphique suivant (Fig. 2a), accompagné de l'estimation de la complexité asymptotique correspondant aux jeux de données, ou la Fig.2b si les données sont déjà triées.

Il est important d'énoncer ces phrases en termes de conjectures à ce niveau de l'étude, afin que l'étudiant garde conscience que seule l'étude mathématique ultérieure pourra éventuellement les prouver.

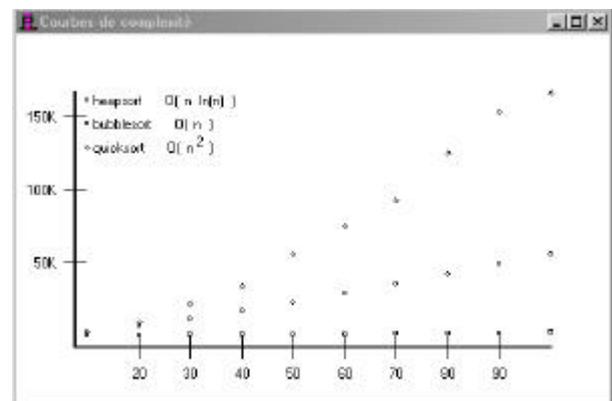


Fig. 2b Complexité d'algorithmes de tris sur des données déjà triées