Atelier CORFEM 2019

La complexité algorithmique : on débranche et on dénombre.

Contributeur
Basile Sauvage, Université de Strasbourg, IREM de Strasbourg, sauvage@unistra.fr

Résumé
Cet atelier a un triple objectif : découvrir des activités d'informatique débranchée afférentes aux algorithmes ; analyser ces activités au regard du concept informatique de complexité algorithmique ; débattre des liens avec le programme de mathématiques au lycée. Dans la majeure partie de l'atelier, les participants expérimenteront eux-mêmes des activités d'informatique débranchée exploitables avec les élèves (à partir du cycle 4), desquelles nous feront émerger la notion de complexité algorithmique. Nous verrons qu'elle s'apparente à plusieurs notions mathématiques : le dénombrement, les suites, les fonctions, les limites, le logarithme et l'exponentielle. L'atelier se terminera par un débat sur l'articulation des notions informatiques et mathématiques, ainsi que de l'opportunité pédagogique d'un éclairage croisé.

Déroulement de l'atelier

Introduction.
Présentation de l'intervenant, des objectifs et de l'organisation de l'atelier. Les participants sont répartis en groupes de 4 sur des ilôts.

Activité : recherche du minimum.
Consigne : 8 cartes sont placées face cachée ; vous avez le droit de retourner deux cartes à la fois pour les comparer ; proposez un algorithme pour trouver la carte de valeur minimale ; comptez le nombre de comparaisons effectuées.
On trouve 7 comparaisons pour 8 cartes, que l'on généralise à (n-1) comparaisons pour n cartes. Nous définissons informellement la notion de complexité T(n) comme le nombre d'opérations en fonction de la « taille du problème », en l'occurrence la taille des données (n).

Activité : tri par sélection.
8 cartes sont placées face cachée. Nous concevons l'algorithme de tri par sélection. Nous dénombrons le nombre de comparaisons : n(n-1)/2 comparaison pour n cartes.

Réflexion sur la complexité asymptotique.
Nous menons une réflexion sur le souhait de se doter d'outils pour comparer les complexités, se débarrasser des termes négligeables (constante additive, n devant n^2, etc.) et de la constante multiplicative dans l'expression de la complexité.

Pour se faire une idée du comportement asymptotique, nous imaginons le scénario où l'on cherche un nom mal orthographié dans un annuaire de n noms. Le premier algorithme renvoie le nom le plus proche (apparenté à une recherche du minimum en complexité linéaire). Le second algorithme trie l'annuaire par proximité décroissante (tri sélection en complexité quadratique). En supposant une machine qui réalise 10^9 comparaisons par seconde, nous construisons le tableau suivant, et commentons l'intérêt prédictif de la complexité asymptotique.

n	T(n) = n	temps	T(n)=n^2	temps
10^3 (annuaire individuel)	10^3	10^-6 sec.	10^6	10^-3 sec.
10^6 (annuaire national)	10^6	10^-3 sec.	10^12	17 minutes
10^9 (annuaire mondial)	10^9	1 sec.	10^18	32 ans

Activité : qui-est-ce ?
Nous réalisons l'activité débranchée imitant le jeu du qui-est-ce, à l'aide de plusieurs arbres de décision. Nous dénombrons le nombre de questions posées. Nous illustrons les notions de complexité au meilleur, au pire, et en moyenne. Dans le cas d'un arbre équilibré, nous illustrons le logarithme en base 2, qui est la profondeur de l'arbre (i.e. le nombre de questions) par rapport au nombre de feuilles (i.e. le nombre de personnages).

Débat.
La complexité algorithmique est en lien avec différentes notions de mathématiques : les suites, les suites récurrentes, les limites, les fonctions, le dénombrement, le logarithme et l'exponentielle. Nous débattons autour de deux questions : Comment mobiliser les mathématiques en informatique ? Comment mobiliser l'informatique en mathématiques ?