Big Data, Big O
et la complexité des algorithmes

Sergey Kirgizov

ESIREM

La complexité des algorithmes

La notation grand O de Bachmann-Landau et autres méthodes de la comparaison asymptotique.

On dit

$$f(x) \in O\big(g(x)\big)$$

lorsqu'il existe des constantes $N > 0$ et $C > 0$ telles que pour tout

$$x > N$$

on a

$$|f(x)| \leq C|g(x)|.$$

Bachmann et Landau

Paul Bachmann
“Analytische
Zahlentheorie“
[Analytic Number Theory]
(in German, 1894).

Edmund Landau
“Handbuch der Lehre von der Verteilung der
Primzahlen”
[Handbook on the theory of the distribution of the
primes]
(in German, 1909).

“The big-O originally stands for "order of" ("Ordnung")”

Donald Knuth

Al Acorn (jeu vidéo Pong, fondateur Atari), Donald Knuth, et Steve Wozniak.
Photo : Robert Scoble @ flickr

"The symbol [big-O] was much later on (1976) viewed by Knuth as a capital omicron, probably in reference to his definition of the symbol Omega"
Big Omicron and big Omega and big Theta, (SIGACT News, Apr.-June 1976)
`https://www-cs-faculty.stanford.edu/~knuth/`

Exemples

Exemples

- $10n + 10000 \in O(n)$
- $10n + 10000 \in O(n^2)$
- $1 \in O(1)$
- $8000999 \in O(1)$
- $n * \frac{n}{2} \in O(n^2)$
- $n * \frac{n}{2} \in O(n^9)$
- $n * \frac{n}{2} \notin O(n)$
- Attaque par force brute $\in O(2^n)$

La mise à jour d'une moyenne

Moyenne $= \frac{x_0 + x_1 + \cdots + x_{n-1} + x_n}{n}$.

$n - 1$ additions, une division. Donc la complexité $O(n)$.

La mise à jour d'une moyenne

Moyenne $= \frac{x_0+x_1+\cdots+x_{n-1}+x_n}{n}$.

$n-1$ additions, une division. Donc la complexité $O(n)$.

Maintenant on ajoute une valeur x_{n+1}.
Quelle est la complexité de la mise à jour de la moyenne ?

La mise à jour d'une moyenne

Moyenne $= \frac{x_0+x_1+\cdots+x_{n-1}+x_n}{n}$.

$n-1$ additions, une division. Donc la complexité $O(n)$.

Maintenant on ajoute une valeur x_{n+1}.
Quelle est la complexité de la mise à jour de la moyenne ?

On peut le faire en $O(1)$ operations ?

Oui !

La mise à jour d'une moyenne en $O(1)$ opérations !

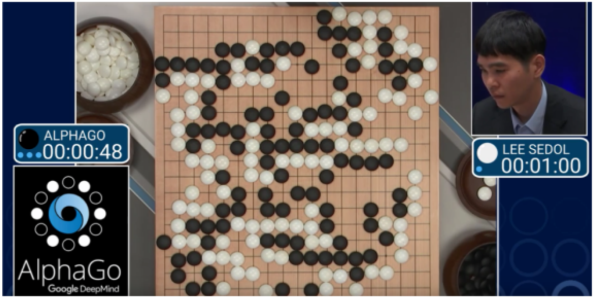
Méthode	Complexité
$M_n = \dfrac{x_1 + x_2 + x_3 + \ldots + x_n}{n}$	$O(n)$
$M_{n+1} = \dfrac{x_1 + x_2 + x_3 + \ldots + x_n + x_{n+1}}{n+1}$	$O(n)$
$M_{n+1} = \dfrac{nM_n + x_{n+1}}{n+1}$	$\boldsymbol{O(1)}$

La moyenne est ces applications

- Apprentissage par renforcement : Q-Learning
  `https://kirgizov.link/teaching/esirem/bigdata-2019/Q-LEARNING.pdf`
- Algorithme du gradient stochastique
- Moyenne glissante, un filtre passe-bas dans le traitement du signal

Apprentissage par renforcement, un exemple

AlphaGo



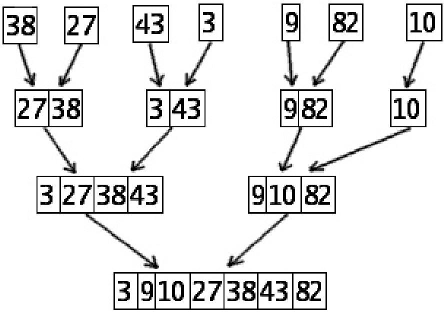
"This program was based on general-purpose AI methods [**including reinforcement learning**], using deep neural networks to mimic expert players, and further improving the program by learning from games played against itself."

— `https://deepmind.com/research/alphago`

Application à l'algorithmique

- Temps de calcul
- Taille de mémoire
- etc...

Exemple d'analyse : tri fusion



- La complexité de la fusion est linéaire
- La complexité totale des fusions de chaque niveau est $O(n)$
- Le nombre des niveaux est $O(\log n)$
- Donc la complexité totale est $O(n \log n)$

Goldstine and John von Neumann, 1945, 1948.

Applications de tri fusion...

“An external merge sort is practical to run using disk or tape drives when the data to be sorted is too large to fit into memory.”

— Wikipédia

IBM 729s de NASA sur le photo

La famille de notations Bachmann–Landau : $O, o, \Theta, \omega, \Omega, \ldots$

Notation	Name[19]	Description	Formal Definition	Limit Definition[20][21][22][19][14]
$f(n) = o(g(n))$	Small O; Small Oh	f is dominated by g asymptotically	$\forall k > 0\ \exists n_0\ \forall n > n_0\ \lvert f(n)\rvert < k \cdot g(n)$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
$f(n) = O(g(n))$	Big O; Big Oh; Big Omicron	$\lvert f\rvert$ is bounded above by g (up to constant factor) asymptotically	$\exists k > 0\ \exists n_0\ \forall n > n_0\ \lvert f(n)\rvert \leq k \cdot g(n)$	$\limsup_{n\to\infty} \frac{\lvert f(n)\rvert}{g(n)} < \infty$
$f(n) = \Theta(g(n))$	Big Theta	f is bounded both above and below by g asymptotically	$\exists k_1 > 0\ \exists k_2 > 0\ \exists n_0\ \forall n > n_0$ $k_1 \cdot g(n) \leq f(n) \leq k_2 \cdot g(n)$	$f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ (Knuth version)
$f(n) \sim g(n)$	On the order of	f is equal to g asymptotically	$\forall \varepsilon > 0\ \exists n_0\ \forall n > n_0\ \left\lvert \frac{f(n)}{g(n)} - 1 \right\rvert < \varepsilon$	$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$
$f(n) = \Omega(g(n))$	Big Omega in number theory (Hardy–Littlewood)	$\lvert f\rvert$ is not dominated by g asymptotically	$\exists k > 0\ \forall n_0\ \exists n > n_0\ \lvert f(n)\rvert \geq k \cdot g(n)$	$\limsup_{n\to\infty} \left\lvert \frac{f(n)}{g(n)} \right\rvert > 0$
$f(n) = \Omega(g(n))$	Big Omega in complexity theory (Knuth)	f is bounded below by g asymptotically	$\exists k > 0\ \exists n_0\ \forall n > n_0\ f(n) \geq k \cdot g(n)$	$\liminf_{n\to\infty} \frac{f(n)}{g(n)} > 0$
$f(n) = \omega(g(n))$	Small Omega	f dominates g asymptotically	$\forall k > 0\ \exists n_0\ \forall n > n_0\ \lvert f(n)\rvert > k \cdot \lvert g(n)\rvert$	$\lim_{n\to\infty} \left\lvert \frac{f(n)}{g(n)} \right\rvert = \infty$

https://en.wikipedia.org/wiki/Big_O_notation

Questions ?