計算量

計算機アルゴリズム特論：2015年度

只木進一

計算量

- アルゴリズムの良さをはかる尺度
  - 計算時間、記憶領域、理解しやすさ、プログラムへの変換の容易さ、等々
- 時間計算量(Time Complexity)
  - どれくらい計算時間が掛るか←講義ではこちらを主に議論
- 領域計算量(Space Complexity)
  - どれくらい記憶領域が必要か

計算量

- 稼働させる計算機に依存
  - CPU、メモリ、ディスクに依存
  - 仮想的計算機で比較
- 入力サイズに依存
  - 入力データのサイズnの関数として評価
  - 漸近的計算量に着目：$n \to$大のときの挙動

- 入力サイズが同じでも差
  - 探索データがリストの先頭なら速いが、末尾なら遅い
  - 平均値、最悪値（最も悪い場合の値）で議論

例：多項式の計算

$$f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$$

- 係数a_iと変数xを与えて、関数の値$f(x)$を求めるのに必要な計算
  - xのべきを求めるための乗算6回
  - 係数とxのべきとの乗算4回
  - 加算4回
  - 合計14回

- Hornerの方法

$$f(x) = \left(\left(\left(\left(a_4 x + a_3\right)x + a_2\right)x + a_1\right)x + a_0\right)$$

- 乗算4回＋加算4回

例：多項式の計算：一般化

- 律儀な方法
  - n次の項： n回の乗算
  - n回の加算

$$P(n)=\sum_{k=0}^{n}k+n=\frac{n(n+1)}{2}+n=\frac{n(n+3)}{2}$$

- Hornerの方法
  - 一つの括弧を閉じる：乗算と加算が各1回

$$P(n) = \sum_{k=0}^{n} 2 = 2n$$

例：逐次探索

10	15	30	37	50	70	81	91

- リストの長さN
- 要素中の一つを探索
  - 位置iにある場合
    - 存在確率$1/N$
    - 比較回数$i+1$

$$\frac{1}{N}\sum_{k=1}^{N} k = \frac{N+1}{2}$$

計算量の評価

- データのサイズnが大きいときに、計算量や計算時間が nに対してどのように増大するか
  - 定数倍にはあまり関心はない
  - nに対して最も速く増大する項に関心がある

オーダー記法

- 二つの関数$f(n)$と$g(n)$を考える

$$0 \le \lim_{n \to \infty} \left| \frac{f(n)}{g(n)} \right| \le c$$

このとき

$$f(n) \in O\left(g(n)\right)$$

- nが大きいとき、$f(n)$は$g(n)$より増加が遅い
  - 通常は、$g(n)$は簡単な関数形

他のオーダー記法

記法	定義
$f(n) \in O(g(n))$	$\lim_{n\to\infty} \left\lvert \frac{f(n)}{g(n)} \right\rvert \leq c$
$f(n) \in o(g(n))$	$\lim_{n\to\infty} \left\lvert \frac{f(n)}{g(n)} \right\rvert = 0$
$f(n) \in \Omega(g(n))$	$\lim_{n\to\infty} \left\lvert \frac{f(n)}{g(n)} \right\rvert \geq c$
$f(n) \in \omega(g(n))$	$\lim_{n\to\infty} \left\lvert \frac{f(n)}{g(n)} \right\rvert = \infty$
$f(n) \in \Theta(g(n))$	$f(n) = O\big(g(n)\big) \wedge$ $f(n) = \Omega(g(n))$

オーダーの関係

$\Omega\left(n^3\right)$

$O\left(n^3\right)$

$3n^3 + 4n^2$

$2n^2 + n$

$3n^4 - n^2$

$n^3 - 4n$

$n \ln n$

$o\left(n^3\right)$

$\Theta\left(n^3\right)$

$\omega\left(n^3\right)$