

離散数理工学 第 10 回
離散確率論：乱択データ構造とアルゴリズム (基礎)

岡本 吉央
okamotoy@uec.ac.jp

電気通信大学

2015 年 1 月 6 日

最終更新：2015 年 1 月 6 日 17:10

スケジュール 前半 (予定)

- | | | |
|---|----------------------|---------|
| 1 | 数え上げの基礎：二項係数と二項定理 | (10/7) |
| ★ | 休講 (体育祭) | (10/14) |
| 2 | 数え上げの基礎：漸化式の立て方 | (10/21) |
| 3 | 数え上げの基礎：漸化式の解き方 (基礎) | (10/28) |
| 4 | 数え上げの基礎：漸化式の解き方 (発展) | (11/4) |
| 5 | 離散代数：群と対称性 | (11/11) |
| 6 | 離散代数：部分群と軌道 | (11/18) |
| 7 | 離散代数：対称性を考慮した数え上げ | (11/25) |

スケジュール 後半 (予定)

- 8 離散確率論：確率の復習と確率不等式 (12/2)
- 9 離散確率論：確率的離散システムの解析 (12/9)
- ★ 中間試験 (12/16)
- 10 離散確率論：乱択データ構造とアルゴリズム (基礎) (1/6)
- 11 離散確率論：乱択データ構造とアルゴリズム (発展) (1/13)
- 12 離散確率論：マルコフ連鎖 (基礎) (1/20)
- ★ 休講 (海外出張) (1/27)
- 13 離散確率論：マルコフ連鎖 (発展) (2/3)
- ★ 期末試験 (2/17?)

注意：予定の変更もありうる

今日の目標

典型的な乱択アルゴリズムの解析ができるようになる

- ▶ 乱択クイックソート

目次

- ① 乱択アルゴリズム
- ② 乱択クイックソート
- ③ 今日のまとめ

乱択アルゴリズム

乱択アルゴリズムとは？

乱数を用いる (あるいは, 用いてもよい) アルゴリズムのこと

確率的アルゴリズム, 乱数使用アルゴリズムとも呼ばれる

なぜ乱数を用いるのか？

- ▶ アルゴリズムを設計しやすくなる
- ▶ アルゴリズムを解析しやすくなる
- ▶ 乱数を使わないとできないことが, 乱数を使うとできる

乱択アルゴリズムの2つの側面

乱択アルゴリズムは乱数を使ってもよいので、振る舞いが確率的になる

類型その1：モンテカルロ・アルゴリズム

- ▶ 実行時間は乱数によって変化しない
- ▶ 出力の正しさが乱数によって変化する（「正しさ」が確率変数）

注：「モンテカルロ法」は違う概念を指す名称なので注意

類型その2：ラスベガス・アルゴリズム

- ▶ 実行時間が乱数によって変化する（「実行時間」が確率変数）
- ▶ 出力の正しさは乱数によって変化しない（つまり、常に正しい）

目次

- ① 乱択アルゴリズム
- ② 乱択クイックソート
- ③ 今日のまとめ

ソートिंग

ソートिंग (整列問題) とは？

- ▶ 入力：異なる n 個の数から成る配列 $A = (a_1, a_2, \dots, a_n)$ (配列)
- ▶ 出力： A の並べ替え $A' = (a'_1, a'_2, \dots, a'_n)$ で,
 $a'_1 < a'_2 < \dots < a'_n$ を満たすもの

例： $A = (8, 3, 5, 1, 7, 9, 2, 4) \rightsquigarrow A' = (1, 2, 3, 4, 5, 7, 8, 9)$

アルゴリズムにおける基本的な問題

クイックソート：基本的な考え方

クイックソート

再帰によってソーティングを行うアルゴリズム (の1つ)

- 1 A から要素を1つ選択 (その要素を **ピボット** と呼ぶ)
 - 2 A を3つの部分に分割
 - ▶ A_1 : ピボットよりも小さい要素から成る配列
 - ▶ p : ピボット
 - ▶ A_2 : ピボットよりも大きい要素から成る配列
 - 3 A_1 と A_2 を再帰的に整列 (結果をそれぞれ A'_1, A'_2 とする)
 - 4 A'_1 と p と A'_2 をこの順に連結して出力
- ▶ アルゴリズムの正当性は直ちに分かる
 - ▶ ピボットの選択法, A_1, A_2 の作成法によって, 細かな実装が変わる

クイックソート：ピボット選択法

ピボットの選択法, A_1, A_2 の作成法によって, 細かな実装が変わる

よく使われるピボット選択法

- ▶ 配列の先頭の要素をピボットとする
- ▶ 配列の先頭の3要素の中央値をピボットとする
- ▶ 配列の中のランダムな要素をピボットとする (乱択アルゴリズム)
(各要素が選択される確率は同一 (一様分布に従う標本抽出))

乱択クイックソート

乱択クイックソート (適当な疑似コード)

```
1: def quicksort(A) # A: array of distinct numbers
2:   return nil if length(A) == 0
3:   p = a number in A chosen uniformly at random
4:   delete p from A
5:   foreach e in A {
6:     print "G"
7:     if e < p then add e to A1 else add e to A2
8:   }
9:   return quicksort(A1) + p + quicksort(A2)
10: end
```

乱択クイックソート

乱択クイックソート (Ruby)

```
1: def quicksort(a)
2:   return nil.to_a if a.length == 0
3:   p = a.sample()
4:   a.delete(p)
4':  a1 = Array.new(); a2 = Array.new()
5:   a.each { |e|
6:     print "G"
7:     e < p ? a1 << e : a2 << e
8:   }
9:   return quicksort(a1) + [p] + quicksort(a2)
10: end
```

ソーティング・アルゴリズムの理論的性能評価

評価尺度として、以下のものがよく用いられる

- ▶ 比較回数：2要素の比較を行った回数
- ▶ 移動回数：要素を移動した回数
- ▶ 領域量：入力配列以外に用いた変数の数

ここでは、**比較回数**に注目 (比較回数 = 出力されたGの個数)

- ▶ 乱択クイックソートにおいて、
比較回数は使用される乱数によって変わる (つまり、確率変数)

X_A = 入力 A に対する乱択クイックソートの比較回数 (確率変数)

乱択クイックソートの解析：注目する確率変数

X_A = 入力 A に対する乱択クイックソートの比較回数 (確率変数)

X_n = $\max\{X_A \mid |A| = n\}$ (確率変数)

- ▶ X_n が表すのは最悪時の比較回数を表す確率変数

目標

X_n が小さいこと (具体的には高確率で $O(n \log n)$ になること)

まず, $E[X_n]$ を考えてみる

- ▶ $E[X_0] = 0$

乱択クイックソートの解析：漸化式

$n \geq 1$ のとき, $X_n = X_A$ となるような入力 A を考えてみると

$$\begin{aligned} E[X_n] &= E[X_A] \\ &= \sum_{i=1}^n E[X_A \mid a'_i \text{ がピボット}] \Pr[a'_i \text{ がピボット}] \\ &\quad (\text{ただし, } a'_i \text{ は } A \text{ の中で } i \text{ 番目に小さい要素}) \\ &= \sum_{i=1}^n E[X_A \mid a'_i \text{ がピボット}] \frac{1}{n} \end{aligned}$$

ここで, a'_i がピボットであるとき

- ▶ $|A_1| = i - 1$, $|A_2| = n - i$
- ▶ $\therefore X_{A_1} \leq X_{i-1}$ かつ $X_{A_2} \leq X_{n-i}$

したがって,

$$E[X_n] \leq \sum_{i=1}^n (n - 1 + E[X_{i-1}] + E[X_{n-i}]) \frac{1}{n} = n - 1 + \sum_{i=0}^{n-1} \frac{2}{n} E[X_i]$$

乱択クイックソートの解析：漸化式 (2)

$E[X_n]$ に関して得られた漸化式

$$E[X_0] = 0$$

$$E[X_n] \leq n - 1 + \sum_{i=0}^{n-1} \frac{2}{n} E[X_i] \quad (n \geq 1)$$

ここで、次の漸化式を満たす数列 $\{t_n\}_{n \geq 0}$ を考える

$$t_0 = 0$$

$$t_n = n - 1 + \sum_{i=0}^{n-1} \frac{2}{n} t_i \quad (n \geq 1)$$

- ▶ このとき、任意の $n \geq 0$ に対して

$$E[X_n] \leq t_n$$

となる (演習問題)

- ▶ つまり、 t_n の上界が分かれば、 $E[X_n]$ の上界となる

漸化式を解く (1)

$$t_n = n - 1 + \sum_{i=0}^{n-1} \frac{2}{n} t_i \quad (n \geq 1)$$

第2式において、添え字をずらしたものを考えると

$$t_{n+1} = n + \sum_{i=0}^n \frac{2}{n+1} t_i \quad (n \geq 0)$$

第2式と第3式を変形すると

$$\begin{aligned} nt_n &= (n-1)n + \sum_{i=0}^{n-1} 2t_i \quad (n \geq 1) \\ (n+1)t_{n+1} &= n(n+1) + \sum_{i=0}^n 2t_i \quad (n \geq 0) \end{aligned}$$

下から上を引くと、 $n \geq 1$ のとき、

$$(n+1)t_{n+1} - nt_n = 2n + 2t_n$$

漸化式を解く (2)

整理すると, $n \geq 1$ のとき,

$$(n+1)t_{n+1} = 2n + (n+2)t_n$$

両辺を $2(n+1)(n+2)$ で割ると, $n \geq 1$ のとき,

$$\frac{t_{n+1}}{2(n+2)} = \frac{n}{(n+1)(n+2)} + \frac{t_n}{2(n+1)}$$

ここで, $s_n = \frac{t_n}{2(n+1)}$ と置くと, 得られる漸化式は

$$s_0 = \frac{t_0}{2(0+1)} = 0$$

$$s_1 = \frac{t_1}{2(1+1)} = 0$$

$$s_{n+1} = \frac{2}{n+2} - \frac{1}{n+1} + s_n \quad (n \geq 1)$$

解けそうな形に近づいてきた

漸化式を解く (3)

$n \geq 2$ のとき,

$$\begin{aligned}
 s_n &= \frac{2}{n+1} - \frac{1}{n} + s_{n-1} \\
 &= \left(\frac{2}{n+1} - \frac{1}{n} \right) + \left(\frac{2}{n} - \frac{1}{n-1} \right) + s_{n-2} \\
 &= \left(\frac{2}{n+1} - \frac{1}{n} \right) + \left(\frac{2}{n} - \frac{1}{n-1} \right) + \cdots + \left(\frac{2}{3} - \frac{1}{2} \right) + s_1 \\
 &= \frac{2}{n+1} + \frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{3} - \frac{1}{2} \\
 &= \frac{1}{n+1} + H_{n+1} - \frac{1}{2} - 1 - \frac{1}{2} \\
 &= H_{n+1} + \frac{1}{n+1} - 2
 \end{aligned}$$

復習: $H_{n+1} = 1 + \frac{1}{2} + \cdots + \frac{1}{n+1}$ (調和数)

漸化式を解く (4)

したがって, $n \geq 0$ に対して,

$$s_n = H_{n+1} + \frac{1}{n+1} - 2$$

したがって, $n \geq 0$ に対して,

$$t_n = 2(n+1)s_n = 2(n+1)H_{n+1} + 2 - 4(n+1)$$

したがって,

$$E[X_n] \leq t_n = 2(n+1)H_{n+1} + 2 - 4(n+1)$$

$H_{n+1} \leq 1 + \ln(n+1)$ なので (前回の演習問題)

$$\begin{aligned} E[X_n] &\leq 2(n+1)(1 + \ln(n+1)) + 2 - 4(n+1) \\ &= 2(n+1)\ln(n+1) - 2n = O(n \log n) \end{aligned}$$

乱択クイックソートの解析：まとめ

ここまでで分かったこと

任意の $n \geq 0$ に対して, $E[X_n] \leq 2(n+1) \ln(n+1)$

したがって, マルコフの不等式を適用してみると

$$\begin{aligned} \Pr[X_n \geq 24(n+1) \ln(n+1)] &\leq \frac{E[X_n]}{24(n+1) \ln(n+1)} \\ &\leq \frac{2(n+1) \ln(n+1)}{24(n+1) \ln(n+1)} \\ &= \frac{1}{12} \end{aligned}$$

- ▶ つまり, 比較回数が $24(n+1) \ln(n+1)$ を超える確率は高くない
- ▶ 「チェルノフ上界の技法」を用いると,
 $n \rightarrow \infty$ のとき, この確率が 0 に収束することを証明できる
 (ちょっと面倒で, 他のアイデアも必要なので, 省略)

目次

① 乱択アルゴリズム

② 乱択クイックソート

③ 今日のまとめ

今日の目標

今日の目標

典型的な乱択アルゴリズムの解析ができるようになる

- ▶ 乱択クイックソート

残った時間の使い方

- ▶ 演習問題をやる
 - ▶ 相談推奨 (ひとりでやらない)
- ▶ 質問をする
 - ▶ 教員と TA は巡回
- ▶ 退室時, 小さな紙に感想など書いて提出する ← 重要
 - ▶ 内容は何でも OK
 - ▶ 匿名で OK

目次

① 乱択アルゴリズム

② 乱択クイックソート

③ 今日のまとめ