離散数理工学 第 11 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

2016 年 1 月 12 日

最終更新：2016 年 1 月 14 日　08:58

スケジュール 前半

1. 数え上げの基礎：二項係数と二項定理 (10/6)
- ⋆ 休講 (体育祭) (10/13)
2. 数え上げの基礎：漸化式の立て方 (10/20)
3. 数え上げの基礎：漸化式の解き方 (基礎) (10/27)
- ⋆ 祝日で休み (11/3)
4. 数え上げの基礎：漸化式の解き方 (発展) (11/10)
5. 離散代数：整数と有限体 (11/17)
6. 離散代数：多項式環 (11/24)
7. 離散代数：多項式環による有限体の構成 (12/1)
8. 離散代数：有限体の応用 (12/8)

スケジュール 後半 (予定)

9. 離散確率論：確率の復習と確率不等式 (12/15)
- ⋆ 中間試験 (12/22)
10. 離散確率論：確率的離散システムの解析 (1/5)
11. 離散確率論：乱択データ構造とアルゴリズム (基礎) (1/12)
12. 離散確率論：乱択データ構造とアルゴリズム (発展) (1/19)
13. 離散確率論：マルコフ連鎖 (基礎) (1/26)
14. 離散確率論：マルコフ連鎖 (発展) (2/2)
- ⋆ 予備日 (2/9)
- ⋆ 期末試験 (2/16?)

注意：予定の変更もありうる

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
- 前進問題
- 乱択クイックソート

目次

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

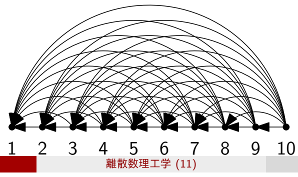
前進問題

前進問題

設定
- 頂点集合を $\{1, 2, \ldots, n\}$ とする有向グラフ $(n \geq 2)$
- 大きな数から小さな数へ向かう辺が必ず存在

行うこと
- 頂点 n から始めて，辺をたどることで頂点 1 に到達

問題
- 辺をいくつたどれば頂点 1 に到達できるか？

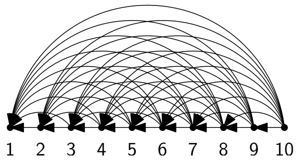


単純なアルゴリズム 1 (乱数を使わない)

1. たどる辺を任意に選び，辺の先に移動する
2. 移動先から出る辺がなければ終了，そうでなければ 1 に戻る

たどる辺の数
- 最悪の場合：$n - 1$ 個
- (最善の場合：1 個)

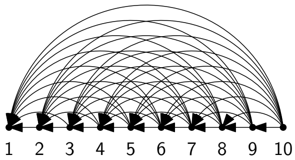
最悪計算量の意味では，よくないアルゴリズム



単純なアルゴリズム 2 (乱択)

1. たどる辺を一様分布に従って選び，辺の先に移動する
2. 移動先から出る辺がなければ終了，そうでなければ 1 に戻る

一様分布に 従って選ぶ	≡	出る辺が k 個ある場合， それぞれを確率 $1/k$ で選ぶ



たどる辺の数の期待値

証明すること

単純なアルゴリズム 2 がたどる辺の数の期待値は H_{n-1}

復習：H_{n-1} は $n-1$ 次調和数であり，

$$H_{n-1} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n-1}$$

事実として，$H_{n-1} = \ln n + O(1)$ が成り立つ
- つまり，単純なアルゴリズム 2 がたどる辺の数の期待値は $O(\log n)$

証明 (1)

- 任意の $k \in \{1, 2, \ldots, n\}$ に対して

$$R_k \quad = \quad \text{頂点 } k \text{ から始めて，頂点 1 への到達までにたどる辺数}$$

とする
- このとき，$\mathrm{E}[R_1] = 0$ で，$k \geq 2$ のとき，

$$\begin{aligned}
\mathrm{E}[R_k] &= \sum_{i=1}^{k-1} \mathrm{E}[R_k \mid \text{頂点 } k \text{ から頂点 } i \text{ に向かう辺を選ぶ}] \\
&\qquad \cdot \Pr(\text{頂点 } k \text{ から頂点 } i \text{ に向かう辺を選ぶ}) \\
&= \sum_{i=1}^{k-1} (\mathrm{E}[1 + R_i]) \cdot \frac{1}{k-1} = \sum_{i=1}^{k-1} (1 + \mathrm{E}[R_i]) \cdot \frac{1}{k-1} \\
&= 1 + \frac{1}{k-1} \sum_{i=1}^{k-1} \mathrm{E}[R_i]
\end{aligned}$$

- この再帰式を解きたい

証明 (2)

- 両辺を $k-1$ 倍すると，$k \geq 2$ のとき

$$(k-1)\mathrm{E}[R_k] = k - 1 + \sum_{i=1}^{k-1} \mathrm{E}[R_i]$$

- よって，$k \geq 3$ のとき

$$(k-2)\mathrm{E}[R_{k-1}] = k - 2 + \sum_{i=1}^{k-2} \mathrm{E}[R_i]$$

- 上の式から下の式を引くと，$k \geq 3$ のとき

$$\begin{aligned}
(k-1)\mathrm{E}[R_k] - (k-2)\mathrm{E}[R_{k-1}] &= 1 + \mathrm{E}[R_{k-1}] \\
(k-1)\mathrm{E}[R_k] - (k-1)\mathrm{E}[R_{k-1}] &= 1 \\
\mathrm{E}[R_k] &= \frac{1}{k-1} + \mathrm{E}[R_{k-1}]
\end{aligned}$$

証明 (3)

- したがって，$k \geq 3$ のとき

$$\begin{aligned}
\mathrm{E}[R_k] &= \frac{1}{k-1} + \mathrm{E}[R_{k-1}] \\
&= \frac{1}{k-1} + \frac{1}{k-2} + \mathrm{E}[R_{k-2}] \\
&= \cdots \\
&= \frac{1}{k-1} + \frac{1}{k-2} + \cdots \frac{1}{2} + \underbrace{\mathrm{E}[R_2]}_{=1} \\
&= H_{k-1}
\end{aligned}$$

- 特に，$n \geq 2$ に対して，

$$\mathrm{E}[R_n] = H_{n-1}$$

□

期待値が分かるとなぜよいか？

- たどる辺数の期待値が分かったからといって，アルゴリズムがそれだけの辺数しかたどらないとは限らない (乱数を使っているから)
- しかし，マルコフの不等式から

$$\begin{aligned}
\Pr(R_n \geq 2H_{n-1}) &\leq \frac{\mathrm{E}[R_n]}{2H_{n-1}} \\
&= \frac{H_{n-1}}{2H_{n-1}} = \frac{1}{2}
\end{aligned}$$

- つまり，$\Pr(R_n < 2H_{n-1}) \geq 1 - \frac{1}{2} = \frac{1}{2}$
- $\frac{1}{2}$ 以上の確率で，たどる辺数は少ない ($2H_{n-1}$ 未満)

しっかりとした確率を導出するために，チェルノフ上界の技法を使う

前進問題：チェルノフ上界の技法

- R_n の代わりに 2^{R_n} を考えてみる

証明したいこと

任意の $k = 1, 2, \ldots, n$ に対して
$$E[2^{R_k}] = k$$

- すなわち，

$$\begin{aligned}
\Pr(R_n \geq 2\log_2 n) &= \Pr(2^{R_n} \geq 2^{2\log_2 n}) \\
&= \Pr(2^{R_n} \geq n^2) \\
&\leq \frac{E[2^{R_n}]}{n^2} = \frac{1}{n}
\end{aligned}$$

- つまり，$1 - \frac{1}{n}$ 以上の確率でたどる辺数は少ない ($2\log_2 n$ 未満)

$$\Pr(R_n < 2\log_2 n) \geq 1 - \frac{1}{n}$$

前進問題：チェルノフ上界の技法 — 証明 (1)

先ほどと同様な手順で進める

- $E[2^{R_1}] = 2^0 = 1$, $E[2^{R_2}] = 2^1 = 2$ で，$k \geq 2$ のとき，

$$\begin{aligned}
E[2^{R_k}] &= \sum_{i=1}^{k-1} E[2^{R_k} \mid \text{頂点 } k \text{ から頂点 } i \text{ に向かう辺を選ぶ}] \\
&\qquad \cdot \Pr(\text{頂点 } k \text{ から頂点 } i \text{ に向かう辺を選ぶ}) \\
&= \sum_{i=1}^{k-1} E[2^{1+R_i}] \cdot \frac{1}{k-1} = \sum_{i=1}^{k-1} E[2 \cdot 2^{R_i}] \cdot \frac{1}{k-1} \\
&= \sum_{i=1}^{k-1} 2\, E[2^{R_i}] \cdot \frac{1}{k-1} = \frac{2}{k-1} \sum_{i=1}^{k-1} E[2^{R_i}]
\end{aligned}$$

- この再帰式を解きたい

前進問題：チェルノフ上界の技法 — 証明 (2)

- 両辺を $k-1$ 倍すると，$k \geq 2$ のとき

$$(k-1)E[2^{R_k}] = 2\sum_{i=1}^{k-1} E[2^{R_i}]$$

- よって，$k \geq 3$ のとき

$$(k-2)E[2^{R_{k-1}}] = 2\sum_{i=1}^{k-2} E[2^{R_i}]$$

- 上の式から下の式を引くと，$k \geq 3$ のとき

$$\begin{aligned}
(k-1)E[2^{R_k}] - (k-2)E[2^{R_{k-1}}] &= 2E[2^{R_{k-1}}] \\
E[2^{R_k}] &= \frac{k}{k-1} E[2^{R_{k-1}}]
\end{aligned}$$

前進問題：チェルノフ上界の技法 — 証明 (3)

- したがって，$k \geq 3$ のとき

$$\begin{aligned}
E[2^{R_k}] &= \frac{k}{k-1} E[2^{R_{k-1}}] \\
&= \frac{k}{k-1} \cdot \frac{k-1}{k-2} E[2^{R_{k-2}}] \\
&= \cdots \\
&= \frac{k}{k-1} \cdot \frac{k-1}{k-2} \cdot \cdots \cdot \frac{3}{2} E[2^{R_2}] \\
&= k
\end{aligned}$$

□

前進問題：まとめ

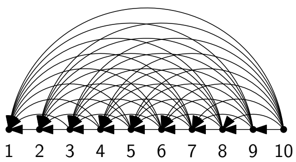
乱数を使わないアルゴリズム

- 最悪時：たどる辺数 $= n - 1$

乱択アルゴリズム

- 期待値：たどる辺数 $= H_{n-1}$ ($= \ln n + O(1)$)
- 高確率：たどる辺数 $= O(\log n)$

∴ 乱数を使うことで，問題を高速に解けた



目次

ソーティング

ソーティング (整列問題) とは？

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

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


乱択クイックソート

乱択クイックソート (Ruby)

```
def quicksort(a)
 return nil.to_a if a.length == 0
 p = a.sample()
 a.delete(p)
 a1 = Array.new(); a2 = Array.new()
 a.each { |e|
   print "G"
   e < p ? a1 << e : a2 << e
 }
 return quicksort(a1) + [p] + quicksort(a2)
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

$$
\begin{aligned}
X_A &= \text{入力 } A \text{ に対する乱択クイックソートの比較回数　(確率変数)}\\
X_n &= \max\{X_A \mid |A| = n\} \quad \text{(確率変数)}
\end{aligned}
$$

- 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，$\mathrm{E}[X_n]$ を考えてみる

- $\mathrm{E}[X_0] = 0$

乱択クイックソートの解析：漸化式 (1)

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

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

乱択クイックソートの解析：漸化式 (2)

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

$$
\mathrm{E}[X_n] = \sum_{i=1}^{n} \mathrm{E}[X_A \mid a'_i \text{ がピボット}] \frac{1}{n}
$$

(ただし，a'_i は A の中で i 番目に小さい要素)

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，$n \geq 1$ のとき

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

乱択クイックソートの解析：漸化式 (3)

$\mathrm{E}[X_n]$ に関して得られた漸化式

$$
\begin{aligned}
\mathrm{E}[X_0] &= 0\\
\mathrm{E}[X_n] &\leq n - 1 + \sum_{i=0}^{n-1} \frac{2}{n} \mathrm{E}[X_i] \quad (n \geq 1)
\end{aligned}
$$

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

$$
\begin{aligned}
t_0 &= 0\\
t_n &= n - 1 + \sum_{i=0}^{n-1} \frac{2}{n} t_i \quad (n \geq 1)
\end{aligned}
$$

- このとき，任意の $n \geq 0$ に対して次が成り立つ (演習問題)
  $$\mathrm{E}[X_n] \leq t_n$$
- つまり，t_n の上界が分かれば，$\mathrm{E}[X_n]$ 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

$$\begin{aligned} s_0 &= \frac{t_0}{2(0+1)} = 0 \\ s_1 &= \frac{t_1}{2(1+1)} = 0 \\ s_{n+1} &= \frac{2}{n+2} - \frac{1}{n+1} + s_n \quad (n \geq 1) \end{aligned}$$

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，

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

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

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

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，$\mathrm{E}[X_n] \leq 2(n+1)\ln(n+1)$

したがって，マルコフの不等式を適用してみると

$$\begin{aligned} \Pr[X_n \geq 4(n+1)\ln(n+1)] &\leq \frac{\mathrm{E}[X_n]}{4(n+1)\ln(n+1)} \\ &\leq \frac{2(n+1)\ln(n+1)}{4(n+1)\ln(n+1)} \\ &= \frac{1}{2} \end{aligned}$$

- つまり，比較回数が $4(n+1)\ln(n+1)$ 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，
  $n \to \infty$ 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，他のアイディアも必要なので，省略)

目次

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

- 前進問題
- 乱択クイックソート