データ構造と アルゴリズム I

第4回

予定 (I)

4/14: 第1回:イントロダクション

4/21: 第2回:アルゴリズムの解析

4/28: 第3回:関数の増加

5/2 (火，金曜日の講義日):
第4回:漸化式，確率論的解析

5/12: 小テスト

5/19: 第5回:乱択アルゴリズム

5/26: 第6回:ヒープソート

6/2: 前半まとめテスト

3.2 標準的な記号と一般的な関数

- 関数 $f(n)$ が単調増加 (monotonically increasing)
$\Leftrightarrow$ $m \le n$ なら $f(m) \le f(n)$
- 関数$f(n)$が強い意味で単調増加(strictly increasing)
$\Leftrightarrow$ $m < n$ なら $f(m) < f(n)$
- 任意の実数 x に対し
  - x 以下の最大の整数を$\lfloor x \rfloor$と表す (x のフロア)
  - x 以上の最小の整数を$\lceil x \rceil$と表す (x のシーリング)
  - $x-1 < \lfloor x \rfloor \le x \le \lceil x \rceil < x+1$
  - C/C++だと，int xで，x/2はxのフロアを返す

多項式

- 次数 d (正整数) の n に関する多項式
(polynomial in n of degree d)

$$p(n) = \sum_{i=0}^{d} a_i n^i$$

- $f(n) = n^{O(1)}$ のとき, 関数 $f(n)$ は多項式的に限定されている (polynomially bounded) という.
  - ある定数 k に対して $f(n) = O(n^k)$ であることと等しい

指数関数

- 全ての実数 a>1 と b に対し

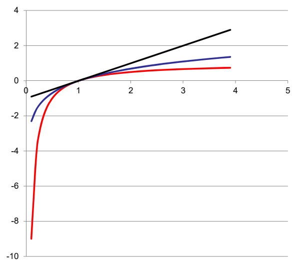
$$\lim_{n\to\infty} \frac{n^b}{a^n} = 0$$

- つまり，底が1より大きな任意の正の指数関数は，どんな多項式よりも早く増加する.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots = \sum_{i=0}^{\infty} \frac{x^i}{i!} \qquad (e = 2.71828\cdots)$$

対数関数

- 以下の記号を用いる
  - $\lg n = \log_2 n$ (2進
    - 計算機科学および こ
  - $\ln n = \log_e n$ (自
  - $\lg^k n = (\lg n)^k$ (べ
  - $\lg \lg n = \lg (\lg n)$ (合
- $\log_b n = O(\lg n)$



$|x| < 1$ のとき $\ln(1+x) = x$

$x > 0$ のとき $1 - \frac{1}{x} \le \ln x \le x - 1$

- $f(n) = \lg^{O(1)} n$ のとき, 関数 $f(n)$ は多項式対数関数的に限定されている (polylogarithmically bounded) という.
- 任意の正の多項式関数はどんな多項式対数関数よりも早く増加する.

階乗

$$n! = \begin{cases} 1 & n = 0 \\ n \cdot (n-1)! & n > 0 \end{cases}$$

- $n! \le n^n$
- スターリングの近似公式

$$n! = \sqrt{2\pi n}\left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$$

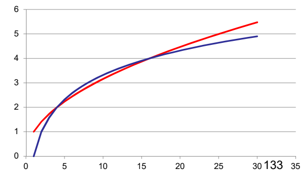
- $\lg(n!) = \Theta(n \lg n)$

課題

- 以下の関数を増加のオーダが小さい順に並べよ

$$n,\ \lg n,\ 2^n, \sqrt{n},\ n!$$

- 大体の予想を
で小さい方を
にして0に収束
- 対数を取るの



課題

$\sum_{k=0}^{n} 3^k = O(3^n)$ を証明せよ

方針：数学的帰納法を用いて,

ある定数 c に関して $\sum_{k=0}^{n} 3^k \le c3^n$ を証明する

$\sum_{k=0}^{n} 3^k = O(3^n)$ を証明

ある定数 c に対して $\sum_{k=0}^{n} 3^k \le c3^n$ を証明する

$n = 0$ のとき, $c \ge 1$ に対して $\sum_{k=0}^{0} 3^k = 1 \le c \cdot 1$

ある定数 c に対して n のときに成り立つと仮定し, n+1 についても成り立つことを示す

$$\begin{aligned}\sum_{k=0}^{n+1} 3^k &= \sum_{k=0}^{n} 3^k + 3^{n+1} \\ &\le c3^n + 3^{n+1} \\ &= \left(\frac{1}{3} + \frac{1}{c}\right) c3^{n+1} \\ &\le c3^{n+1}\end{aligned}$$

$(1/3+1/c) \le 1$ つまり
$c \ge 3/2$ のとき成り立つ

数学的帰納法の応用

- 50組の夫婦がいる村がある.
- その村の夫は全員浮気している.
- 村の女は自分の夫以外の男が浮気していれば即座にそれが分かる. しかし、自分の夫が浮気していてもそれを直接知ることができない.
- 村にある掟により, 自分の夫が浮気していることを証明できる女は, その夫をその日のうちに殺さなければならない.
- ある日, 決して過ちを犯さないことで知られている女王がこの村を訪れ, 少なくとも一人の夫が浮気していると宣言した.
- この村に何が起こるか?

数学的帰納法の応用(続き)

- (i) 浮気している夫が一人だけ (h1) の場合,
  - その妻 (w1) は, 他の夫が浮気していない事実を知っているため, 自分の夫が浮気していることが分かる → その日のうちにh1は殺される
- (ii) 浮気している夫が二人 (h1, h2) の場合,
  - その妻 (w1, w2) は, (i)の状況を想定
  - w1は翌朝になって驚く (w2は何を考えている?)
  - 唯一の可能性はh1が浮気していること→二日目にh1 (およびh2) は殺される
- 以下同様, 50日目にすべての夫が殺される

137

この問題の面白さ／気持ち悪さ

- 女王の発言は, 何も新しい情報を加えていない (newsではない)
- この村に浮気している夫がいることは, 妻の間では周知の事実:すべての妻が(自分の夫を除く)49人の浮気している夫がいることを知っている
- 誰もが知っている事実, かつ, この事実を誰もが知っているということを誰もが知っていることを, 改めて確認すると知識が増える?
- 浮気している夫が一人だけという仮想的な世界では, 女王の発言は, 唯一の浮気している夫の妻に関してはnews
- このことが, 浮気している夫が二人だけの世界に影響を与える
- 以下同様に, 50人の世界に影響が及んでいる

138

もう一つの例題(死刑囚の推論)

- 王様がある囚人を死刑にすることにした.
- 今日は日曜日で, 月曜日から金曜日のいずれかに死刑を執行する必要がある.
- ただし, 死刑の日付は死刑囚には分からないようにしなければならない(その日の朝に, 死刑が執行されることが死刑囚に分かってはいけない)
- いつなら死刑を執行できるか？

139

死刑囚の推論(続き)

- 死刑囚は考えた...
- 金曜日には死刑はない. なぜなら, 金曜日の朝になれば, 確実に死刑があることが分かるから.
- そうすると, 木曜日でもない. 金曜日でないのだから, 木曜日の朝になれば, 確実に死刑になることが分かる.
- 以下同様に, どの日付でも死刑はできない.
- やった！ 死刑は実行できない！

140

死刑囚の推論(結末)

- 金曜日に王様は死刑囚に言った. ``本日, お前の死刑を執行する. ‘‘
- 死刑囚は驚いて言った. ``王様, それは何かの間違いです. 私の死刑は執行することは出来ません. ‘‘
- 王様は聞いた.``それは確かかね, お前の死刑を執行することは出来ないのかね？ ‘‘
- 死刑囚は理路整然と説明した.
- 王様は言った.``確かにその通りだ. しかし, お前は今日は死刑ではないと思っていたのだから, お前の首を切っても良いわけだ. ‘’ --- 死刑は執行された...
- 死刑囚の推論が間違ってた? それとも王様が間違ってる?

141

自己言及

- “すべてのクレタ人は嘘つきだ” --- 言ったのはクレタ人のエピメニデス
- “この文は間違っている” --- この文は正しい?
- 自己言及はやっかい
- ゲーデルの論理式: “この論理式は自然数論の公理系では証明できない”という論理式
- 自然数論の公理系が無矛盾である限り, ゲーデルの論理式もその否定も証明できない

142

死刑囚の考えは何がまずかったか？

- ``月曜日から金曜日のいずれかに死刑を執行する必要がある"をp,``死刑の日付は死刑囚には分からないようにしなければならない"をqとする.
- ある日までに死刑がなかったことをrとする.
- ある日の朝に死刑が実行されることが分かることをkとする
- qは, ``p, q, rからkが結論できるならkでない."となる.
- qは自己言及があり, かつp, q, rから矛盾が導き出せる(死刑囚の推論). → 矛盾する前提からはなんでも導き出せる. 死刑をしてもしなくても良い.

143

オーダの正しくない導き方

$\sum_{k=0}^{n} k = O(n)$ を導く（本当は$O(n^2)$）

$\sum_{k=0}^{n} k = O(n)$ を仮定して,

$\sum_{k=0}^{n+1} k = O(n+1)$が言えれば良い！

$$\sum_{k=0}^{n+1} k = \sum_{k=0}^{n} k + (n+1)$$
$$= O(n) + (n+1)$$
$$= O(n+1)$$

証明できた?

144

$\sum_{k=0}^{n} k = O(n)$ の正しい定義：

ある定数cが存在し,

十分大きなすべての nに関して

$\sum_{k=0}^{n} k \le c \cdot n$が成立

$\sum_{k=0}^{n} k \le cn$ とすると

$$\sum_{k=0}^{n+1} k \le cn + (n+1)$$
$$= c(n+1) - c + n + 1$$

不成立!
前の議論は$O(n)$と$O(n+1)$で違う定数cを使っている!

145

4. 分割統治

- マージソートの実行時間

$$T(n) = \begin{cases} \Theta(1) & n = 1\text{のとき} \\ 2T(n/2) + \Theta(n) & n > 1\text{のとき} \end{cases}$$

$T(n) = \Theta(n \lg n)$

- このような漸化式を解く手法
  1. 置き換え法 (substitution method)
  2. 再帰木 (recursion-tree method)
  3. 分類法 (master method) --- こういう形ならこうなるというルールがある（いつでも適用可能とは限らない）

146

4.3 置き換え法

- 解の形を推測して数学的帰納法で示す
- $T(n) = 2T(\lfloor n/2 \rfloor) + n$ の上界を求める
- $T(n) = O(n \lg n)$ と推測する

面倒なので, $n = 2^m$を仮定

$T(2^m) \le c \cdot 2^m \lg 2^m = cm \cdot 2^m$を証明する

$T(2^m) \le cm \cdot 2^m$を仮定して,

$T(2^{m+1}) \le c(m+1)2^{m+1}$を導く

$T(2^{m+1}) \le 2cm \cdot 2^m + 2^{m+1} = [c(m+1) - c + 1]2^{m+1}$

$c \ge 1$で成立

147

- $T(n) \le cn \lg n$ が境界条件 (n=1の場合) についても成り立つことを示す必要がある.
  - $T(1) = 1$ の場合,
  - $c \lg 1 = 0 < T(1)$より成り立たない
- 漸近記法では $n \ge n_0$ に対して成り立てばよい
  - $T(2) = 4$で成り立てばよい
  - $T(2) \le c \cdot 2 \lg 2$
- $c \ge 2$ なら成り立つ

148

課題

- 以下の漸化式 $T(n) = T(n/2) + n$
- 帰納法／置き換え法で次の性質を示せ： $T(n) = O(n)$
  - $O(n)$ の定義に従って忠実に証明すること
    - 十分大きなすべてのnについて，ある定数cが存在し，以下が成立する： $T(n) \le c \cdot n$
  - 境界条件は無視して良い
  - nは2の累乗であることを仮定してよい： $n = 2^m$

mで$T(2^m) \le c \cdot 2^m$が成立すると仮定し，

$T(2^{m+1}) \le c \cdot 2^{m+1}$を示す．

149

証明

$T(n) = T(n/2) + n$

mで$T(2^m) \le c \cdot 2^m$が成立すると仮定し，

$T(2^{m+1}) \le c(2^{m+1})$を示す．

$$T(2^{m+1}) \le c \cdot 2^m + 2^{m+1} = (\frac{1}{2} + \frac{1}{c})c \cdot 2^{m+1} \le c \cdot 2^{m+1}$$

$c \ge 2$で成立

150

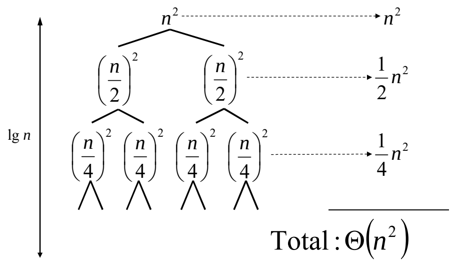
4.4 再帰木

- 漸化式の展開を繰り返し，nと初期条件のみに依存する項の和で表現する
- 上記を視覚化したものが再帰木

151

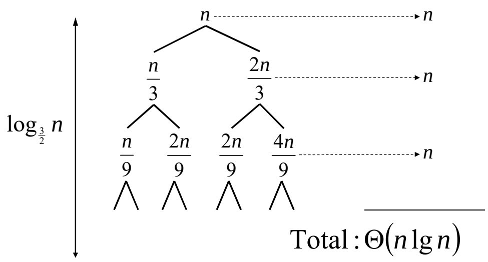
再帰木 (recursion tree)

- 再帰を視覚化するのに便利な方法
- $T(n) = 2T(n/2)+n^2$ の例



152

$T(n) = T(n/3)+T(2n/3)+n$ の例



153

C.2 確率

- 確率は標本空間 S によって定義される．
- S の要素を根元事象という．
- 例：2枚の硬貨を投げる試行の場合
  $S = \{HH, HT, TH, TT\}$
- 事象：標本空間の部分集合
  - 2枚のうち1枚が表である事象は $\{HT, TH\}$
- S に関する確率分布 $\Pr\{S\}$: 事象から実数への写像 --- 事象に対して，事象が生じる確率を与える．

154

確率公理

1. 任意の事象 A に対して $\Pr\{A\}\geq 0$
2. $\Pr\{S\}=1$
3. 互いに素な任意の2つの事象 A, B に対し
   $\Pr\{A\cup B\} = \Pr\{A\} + \Pr\{B\}$

$\Pr\{\varnothing\}=0$

$A\subseteq B$のとき$\Pr\{A\}\leq\Pr\{B\}$

$\Pr\{\overline{A}\}=1-\Pr\{A\}$

$$\begin{aligned}\Pr\{A\cup B\}&=\Pr\{A\}+\Pr\{B\}-\Pr\{A\cap B\}\\&\leq\Pr\{A\}+\Pr\{B\}\end{aligned}$$

155

条件付確率

- 2枚のコインのうち少なくとも1枚が表であることがわかったときに，両方とも表の確率は?
- A: 両方が表である事象
- B: 少なくとも一方が表である事象

$$\Pr\{A\mid B\}=\frac{\Pr\{A\cap B\}}{\Pr\{B\}}\quad(\Pr\{B\}\neq 0\text{ のとき})$$

$$=\frac{1/4}{3/4}=\frac{1}{3}$$

HH	HT
TH	TT

156

一般に　$\Pr\{A\mid B\}\cdot\Pr\{B\}=\Pr\{A\cap B\}$

$\Pr\{A\}\cdot\Pr\{B\}=\Pr\{A\cap B\}$　または

$\Pr\{A\mid B\}=\Pr\{A\}$　のとき

事象 A, B は独立 (independent) であるという.

157

Bayes (ベイズ) の定理

条件付確率の定義より

$$\begin{aligned}\Pr\{A\cap B\}&=\Pr\{B\}\cdot\Pr\{A\mid B\}\\&=\Pr\{A\}\cdot\Pr\{B\mid A\}\end{aligned}$$

$$\begin{aligned}\Pr\{B\}&=\Pr\{B\cap A\}+\Pr\{B\cap\overline{A}\}\\&=\Pr\{A\}\Pr\{B\mid A\}+\Pr\{\overline{A}\}\Pr\{B\mid\overline{A}\}\end{aligned}$$

$$\begin{aligned}\Pr\{A\mid B\}&=\frac{\Pr\{A\}\cdot\Pr\{B\mid A\}}{\Pr\{B\}}\\&=\frac{\Pr\{A\}\cdot\Pr\{B\mid A\}}{\Pr\{A\}\Pr\{B\mid A\}+\Pr\{\overline{A}\}\Pr\{B\mid\overline{A}\}}\end{aligned}$$

158

演習：事後確率

- x: 公正な硬貨 (表の確率が1/2)
- y: インチキな硬貨 (表の確率が1)
- xかyをランダムに選択し，それを2回振る.
- 2回とも表のときに，選んだ硬貨がyである確率は?
  - ベイズの定理を用いて求める
- A: y が選択される事象
- B: 硬貨が2回とも表である事象
  - 求める確率は Pr(A|B)

159

事後確率

- A: y が選択される事象
- B: 硬貨が2回とも表である事象

$$\begin{aligned}\Pr\{A\mid B\}&=\frac{\Pr\{B\mid A\}\Pr\{A\}}{\Pr\{B\mid A\}\Pr\{A\}+\Pr\{B\mid\neg A\}\Pr\{\neg A\}}\\&=\frac{1\cdot(1/2)}{1\cdot(1/2)+(1/4)\cdot(1/2)}\\&=\frac{4}{5}\end{aligned}$$

160

ベイズの定理の使い方

- 病気D (風邪，癌など)にかかる確率：Pr{D}
- ある病気Dになっているとき，ある症状S (熱，咳など)が生じる確率: Pr{S|D}

--- これらは統計的に求めることができる

- これらの確率を使って診断ができる
  - 症状を観察して，実際に病気になっている確率を得る
- Pr{風邪|熱}

＝Pr{熱｜風邪}･Pr{風邪}／(Pr{熱|風邪}･Pr{風邪}+Pr{熱｜癌}･Pr{癌}+…)

161

病気に関する検診

- 非常に珍しい，不治の難病Dにかかっている可能性をチェックするテストがある.
- Pr{D}= 0.0001=1/10000 --- 一万人に一人
- テストの精度は99%：
  - False-positive: 本当は病気ではないのに病気だと診断される: 1%, Pr{T|¬D}=0.01
  - False-negative: 本当は病気なのに，病気ではないと診断される: 1%, Pr{¬T|D}=0.01
- もし，このテストを受けてチェックに引っかかったらどうする?

162

病気に関する検診

- 悲観して自殺する前に，条件付き確率を考えよう!
  - Pr{D|T}
    = Pr{T|D}･Pr{D}／(Pr{T|D}･Pr{D}+Pr{T|¬D}･Pr{¬D})
    = 0.99*0.0001／(0.99*0.0001+0.01*0.9999)≒0.0098…
- 本当に病気の可能性は1%未満
- 直感的には，1万人中，100人はチェックに引っかかっているが，そのうちで本当に病気なのは一人だけ

163