データ構造と アルゴリズムI

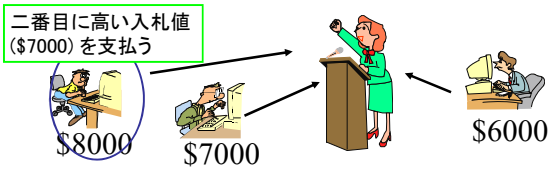
第3回

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

Vickrey入札

- 最高値の入札者が落札
- 支払う金額は二番目に高い入札値



Vickrey入札の性質

- 自分の支払う意思のあるぎりぎりの金額を入札するのが最適 (正直が最良の策).
- 他者の入札値を知っても利益にならない.
  - 自分の評価値が$8000の場合:
  (A) 他者の入札値の最高額が$8000未満:
  何を入札しても支払額は同じ
  (B) $8000以上: 何を入札しても利益を得ることは不可能

全員が（自発的に）正直な入札をして，必ずenvy-freeな割当てが可能

キーワード広告

キーワード広告の概要

- 広告主はキーワードに対して入札額を設定
- キーワードが検索されると，入札額の高い順に広告がユーザに提示される
- ターゲットを絞った広告が可能
- ユーザが広告のリンクをクリックした場合にのみ，広告主はサーチエンジンに広告料を支払う（pay-per-click）
- 広告料をどう設定するか?

広告料の設定方法

- 初期のシステム：支払額は広告主自身の入札額
  - 入札額の設定方法が難しい
  - ダミーの検索を行い，入札額を変化させる行為が蔓延
- k番目のスロットを得た広告主は，k+1番目の入札額に等しい額を払う方式 (第二価格/Vickreyに準ずる方式) に変更
  - 入札額が安定

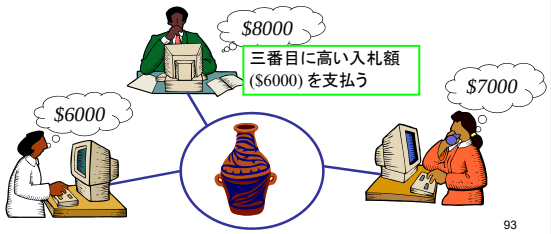
Lessons Learned:

- Vickrey入札は，理論的に優れた性質を持つにも関わらず，従来は広く用いられることはなかった
- 今では世界中で最も頻繁に実行されている入札方式
- 人間が用いるオフラインの取引では問題が表面化しなかったメカニズムでもインターネット上で頻繁に用いられる場合には破綻する可能性がある

92

クイズ：大は小を兼ねる！？

第二価格秘密入札で，勝者の支払い額を2番目に高い入札額ではなく3番目に高い入札額にしたら，正直は最良の策? Envy-free?



93

分割統治法

- 問題の再帰的な (recursive) 構造を利用
  分割：問題をいくつかの小さな部分問題に分割
  統治：各部分問題を再帰的に解く
  統合：それらの解を組合わせて元の問題の解を構成
- マージソートでは
  分割：n 要素の列を $n/2$ 要素の2つの部分列に分割
  統治：マージソートを用いて2つの部分列をソート
  統合：2つのソートされた部分列を統合して答を得る

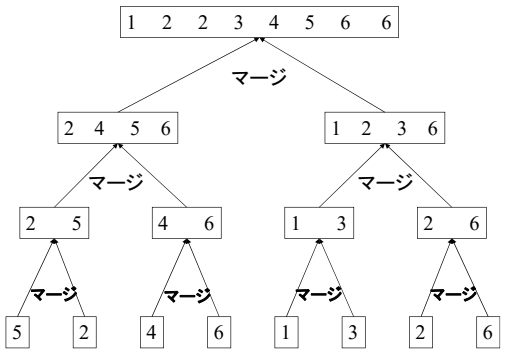
94

マージソート

```
void MERGE-SORT(data A[], int p, int r)    // A[p..r] をソート
{
   int q;
   if (p < r) {                            // p==r ならソートする必要なし
        q = (p+r)/2;
        MERGE-SORT(A, p, q);     // A[p..q] を再帰的にソート
        MERGE-SORT(A, q+1, r); // A[q+1..r] を再帰的にソート
        MERGE(A, p, q, r);
              // ソートされた部分列 A[p..q] と A[q+1..r] を統合
   }
}

main ()
{
  初期化
   MERGE-SORT(D,1,n);
}
```

95



96

マージ

```
#include <limits>
#define infinity  LONG_MAX
void MERGE(data A[], int p, int q, int r)
            // ソートされた部分列 A[p..q] と A[q+1..r] を統合
{ int n, n1, n2, i, j, k;
  n = r-p+1;
  data L[n+1];
  data R[n+1]; // 部分列を保存する一時的な配列
  n1=q-p+1; n2=r-q;
  for (i=1;i<=n1;i++) L[i]=A[p+i-1]; //一時的な配列に保存
  for (i=1;i<=n2;i++) R[i]=A[q+i]; //一時的な配列に保存
  L[n1+1]=infinity; R[n2+1]=infinity; //番兵
                                  注意：i++は副作用としてiの値を1増加
  i = 1; j = 1;                   させるが，返す値はiの増加する前の値.
  for (k=p; k<=r;k++)             ++iだと返す値はiの増加後の値.
    if (L[i] <= R[j]) A[k] =L[i++]; // 前半のほうが小さい
    else A[k] = R[j++];           // 後半のほうが小さい
}
```

97

分割統治アルゴリズムの解析

- 全体の実行時間は問題のサイズに関する漸化式 (recurrence) で記述できることが多い
- サイズ n の問題に関する実行時間を $T(n)$ とする
- $n \le c$ (ある定数) ならば定数時間($\Theta(1)$)とする
- 問題を a 個の部分問題に分割し, それぞれが元のサイズの $1/b$ 倍になったとすると

$$T(n) = \begin{cases} \Theta(1) & n \le c \text{のとき} \\ aT(n/b) + D(n) + C(n) & \text{それ以外} \end{cases}$$

$D(n)$, $C(n)$: 問題の分割, 統合にかかる時間

98

マージソートの解析

- n は2のべき乗と仮定する
- n=1のとき $T(n) = \Theta(1)$
- n>1のとき
  - 分割: $D(n) = \Theta(1)$
  - 統治: 再帰的にサイズ$n/2$の部分問題を解く
    $2T(n/2)$
  - 統合: MERGEは $C(n) = \Theta(n)$

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

$T(n)= \Theta(n \lg n)$ となる (後で証明)

99

アルゴリズムの重要性

- コンピュータが速くても, 実行時間のオーダが大きいアルゴリズムは役に立たない
- スーパーコンピュータで挿入ソートを実行
  - 1秒間に1億命令実行,
  - $2n^2$ 命令必要と仮定
- パーソナルコンピュータでマージソートを実行
  - 1秒間に100万命令実行,
  - 50 $n \lg n$ 命令必要と仮定

100

- 100万個の数の配列のソート
- スーパーコンピュータで挿入ソート

$$\frac{2 \cdot (10^6)^2 \text{命令}}{10^8 \text{命令/秒}} = 20{,}000\text{秒} \approx 5.56\text{時間}$$

- パーソナルコンピュータでマージソート

$$\frac{50 \cdot 10^6 \lg 10^6 \text{命令}}{10^6 \text{命令/秒}} = 1{,}000\text{秒} \approx 16.67\text{分}$$

- オーダの低いアルゴリズムの開発が重要

101

3. 関数の増加

- アルゴリズムの効率を実行時間のオーダで特徴付け, 相対的な比較を行う
- 入力サイズ n が大きいときの挙動を知りたい
- アルゴリズムの漸近的 (asymptotic) な効率を調べる

102

3.1 漸近記号

Θ-記法

- ある関数 $g(n)$ に対し, $\Theta(g(n))$ は次のような性質を持つ関数の集合と定義する

$\Theta(g(n)) = \{f(n)$:ある正の定数c_1, c_2, n_0 が存在し, 全ての $n \ge n_0$に対して $0 \le c_1 g(n) \le f(n) \le c_2 g(n)$ が成立$\}$

$$\Theta(g(n)) = \{ f(n) : \exists c_1 > 0, \exists c_2 > 0, \exists n_0 > 0, \forall n \ge n_0 \quad 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \}$$

- $f(n) = \Theta(g(n))$ は $f(n) \in \Theta(g(n))$ を意味する
- $2n^2+3n+1 = 2n^2 + \Theta(n)$ という表現も用いる

103

限量子の順序

- 順序によって意味が違う

$\forall x \exists y\ loves(x, y)$:

誰にでも愛する人が存在

yはxに依存して異なる値を取ってよい

$\exists y \forall x\ loves(x, y)$:

すべての人から愛されている人が存在

yはxによらない同じ値

104

課題

$\frac{1}{2}n^2 - 3n = \Theta(n^2)$を示せ

方針：全ての$n \geq n_0$に対して

$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$となるような

n_0, c_1, c_2が存在することを示す

(そのような具体的なn_0, c_1, c_2を見つける)

105

全ての$n \geq n_0$に対して$c_1 n^2 \leq \frac{1}{2}n^2 - 3n \leq c_2 n^2$となればよい

n^2で割ると$c_1 \leq \frac{1}{2} - \frac{3}{n} \leq c_2$

$c_2 \geq \frac{1}{2}$なら$n \geq 1$に対して右辺の不等号 が成立

$c_1 \leq \frac{1}{14}$なら$n \geq 7$に対して左辺の不等号 が成立

よって$c_1 = \frac{1}{14}, c_2 = \frac{1}{2}, n_0 = 7$とすれば成立

106

課題

$6n^3 \neq \Theta(n^2)$を示せ.

方針：背理法で示す

全ての$n \geq n_0$に対して

$6n^3 \leq c_2 n^2$であるような

定数c_2, n_0が存在したとして

矛盾を導く.

107

$6n^3 \neq \Theta(n^2)$を背理法で示す

全ての$n \geq n_0$に対して$6n^3 \leq c_2 n^2$であるような

定数c_2, n_0が存在したとする

任意の大きなnに対して$6n \leq c_2$となり矛盾

$f(n) = an^2 + bn + c = \Theta(n^2)$

任意のd次多項式$p(n) = \sum_{i=0}^{d} a_i n^i$に対し$(a_d > 0)$

$p(n) = \Theta(n^d)$

108

O-記法

- ある関数 $g(n)$ に対し, $O(g(n))$ は次のような集合と定義する

$O(g(n)) = \{f(n)$:正の定数c, n_0 が存在し,
全ての $n \geq n_0$に対して
$0 \leq f(n) \leq c\ g(n)$ が成立$\}$

$O(g(n)) = \{f(n) : \exists c > 0, \exists n_0 > 0, \forall n \geq n_0$
$0 \leq f(n) \leq cg(n)\}$

- $f(n) = \Theta(g(n))$ ならば $f(n) = O(g(n))$
- $n = O(n^2)$ も正しい表現

109

- 挿入ソートの実行時間は$O(n^2)$…正解
- 挿入ソートの実行時間は$\Theta(n^2)$… 間違い
  - ソートされた入力に対しては$\Theta(n)$
- 実行時間が$O(n^2)$であるという場合，最悪実行時間について言っている
  - 実行時間は入力データ数のみでなく，入力データそのものに依存する
  - 最悪実行時間はデータ数 n のみに依存

Ω-記法

- ある関数 $g(n)$ に対し，$\Omega(g(n))$ は次のような集合と定義する

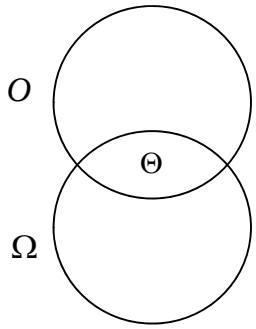
$\Omega(g(n)) = \{f(n)$: ある正の定数c, n_0 が存在し，
全ての $n \geq n_0$に対して
$0 \leq c\, g(n) \leq f(n)$ が成立$\}$

$\Omega(g(n)) = \{ f(n) : \exists c > 0, \exists n_0 > 0\ \forall n \geq n_0$
$0 \leq c\, g(n) \leq f(n)\}$

- $f(n), g(n)$ に対して $f(n) = \Theta(g(n))$ であるための必要十分条件は
  $f(n) = O(g(n))$ かつ $f(n) = \Omega(g(n))$

- Ω-記法は下界を表す
- 挿入ソートの実行時間が$\Omega(n)$とは
  - このアルゴリズムではどのような入力に対しても少なくともn に比例した時間が必要という意味
  - 最良の実行時間について言っている
- アルゴリズムの実行時間が$\Omega(n^2)$とは
  - どのような入力に対しても少なくとも n^2 に比例する時間がかかるという意味

Θ ,O, Ωの関係



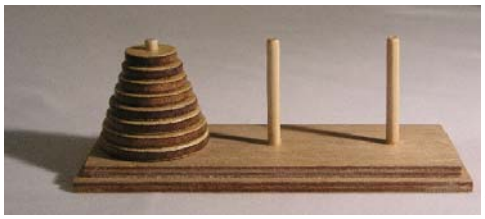
課題：正しいものを選べ

1. 挿入ソートの実行時間は$O(n^2)$
2. 挿入ソートの実行時間は$O(n)$
3. 挿入ソートの実行時間は$\Omega(n^2)$
4. 挿入ソートの実行時間は$\Omega(n)$
5. 挿入ソートの実行時間は$\Theta(n^2)$
6. 挿入ソートの最悪の実行時間は$\Theta(n^2)$
7. 挿入ソートの最良の実行時間は$\Theta(n)$

1,3が両方とも真なら5は真，逆も成立
6が真なら1は真（逆はダメ）
7が真なら4は真（逆はダメ）
2が真なら1は真（逆はダメ）
3が真なら4は真（逆はダメ）
正解は1, 4, 6, 7

分割統治の威力：ハノイの塔

- インドのガンジス河岸辺のベナレスに、世界の中心を表すという巨大な寺院がある。そこには青銅の板の上に、長さ1キュビット、太さが蜂の体ほどの3本のダイヤモンドの針が立てられている。そのうちの1本には、天地創造のときに神が64枚の純金の円盤を大きい円盤から順に重ねて置いた。これが「ブラフマーの塔」である。司祭たちはそこで、昼夜を通して円盤を別の柱に移し替えている。そして、全ての円盤の移し替えが終わったときに、世界は崩壊し終焉を迎える。



ハノイの塔の解法

- 解くべき問題: hanoi(n, A, B, C)
  - n個のAにある円盤をBに移す
- この問題を以下のように分割
  - 上からn-1個の円盤をAからCに移す，すなわち hanoi(n-1, A, C, B) を解く
  - 次に，一番大きい円盤をAからBに移す
  - さらに，Cにあるn-1個の円盤をBに移す．すなわちhanoi(n-1, C, B, A)を解く

116

ハノイの塔の解法

- hanoi(3, A, B, C) を解くためには，まずhanoi(2, A, C, B) を解く必要がある．
- hanoi(2, A, C, B) を解くためには，まずhanoi(1, A, B, C)を解く必要がある --- これは簡単，一番小さい円盤をBに移せばよい．
- 次に，Aにある二番目に大きい円盤をCに移す --- これも簡単
- 次に，hanoi(1, B, C, A)を解く --- Bの最小の円盤をCの二番目の円盤に重ねる
- 次に，Aにある最大の円盤をBに移動
- 以下同様…

117

ハノイの塔の実行手順の解析

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

結局，$T(n)=2^n-1$

118

宝石の分配

問題設定:

- A, B, C, D, Eの5人の海賊が，100個の宝石を分配しようとしている
- Aから順に，分配方法 (誰がいくつ取るか) を提案する．
- 提案された分配方法に対して，提案者も含めて多数決を取る (同数の場合は否決とみなす)
- 可決の場合は提案方法を採用，否決の場合は，提案者を皆で殺して，次の順番の海賊が提案を行う

仮定:

- 自分は死にたくない (全く宝石がもらえなくても死ぬよりはまし)
- よりたくさんの宝石がもらえる方が (他者の生死に関わらず) うれしい
- もらえる宝石の数が同じなら，大勢殺した方がうれしい

Aは何を提案したら生き延びて，より多くの宝石を手に入れられるか？ --- 分割統治／小さい問題から考える

宝石の分配 (続き)

- A, B, Cが殺され，Dが提案する場合を考える．
- EはDのどんな提案も反対し，Dの提案は否決される
- Eが100個手に入れ，効用は (A:-∞, B:-∞, C:-∞, D:-∞, E:100+4ε)

宝石の分配 (続き)

- A, Bが殺され，Cが提案を行う場合を考える．
- EはCのどんな提案も反対する
- DはCのどんな提案でも受け入れる (自分の番になれば死ぬのは確実)
- Cの提案，”俺が全部取る”は，C, Dの賛成で可決，効用は (A:-∞, B:-∞, C:100+2ε, D:0+2ε, E:0+2ε)

宝石の分配 (続き)

- Aが殺され，Bが提案を行う場合を考える.
- CはBのどんな提案も反対する
- D, Eは一個もらえれば満足 (Cの番になれば何ももらえない)
- Bの提案，”俺が98, D, Eは一個ずつ”は，B, D, Eの賛成で可決, 効用は
  (A:-∞, B:98+ε, C:0+ε, D:1+ε, E:1+ε)

宝石の分配 (続き)

- Aが提案を行う場合を考える
- Bに順番が回った場合は
  (A:-∞, B:98+ε, C:0+ε, D:1+ε, E:1+ε)
- 自分以外の二人以上の賛成が得られればよい
- 例えば，”俺が97, Cは1, D (or E) は2”
  なら，可決された場合の利益は，
  (A:97, B:0, C:1, D:2, E:0), よってA, C, Dの賛成多数で可決!

宝石の分配 (続き)

- A以外が共謀すればもう少し利益を増やせるか?
- Cの提案にD, Eが共謀して反対することはない --- 絶対EはDを裏切る
- よってCに回った場合の結果は安定
- 同様に，Bの提案に，C, D, E中の少なくとも二人が共謀して反対することはない
- よってBに回った場合の結果は安定
- 同様に，Aの提案にB, C, D, E中の少なくとも三人が共謀して反対することはない
- (人質を取るとかの) 強制力がないと，安定した共謀は成立しない