

ITプランニング演習 アルゴリズム

情報学部 経営情報学科
堀田敬介

2007/10/30, Tue.

アルゴリズム

- アルゴリズムについて勉強しましょう

アルゴリズム？



例えば, これ?
☆アルゴリズム体操

アルゴリズム

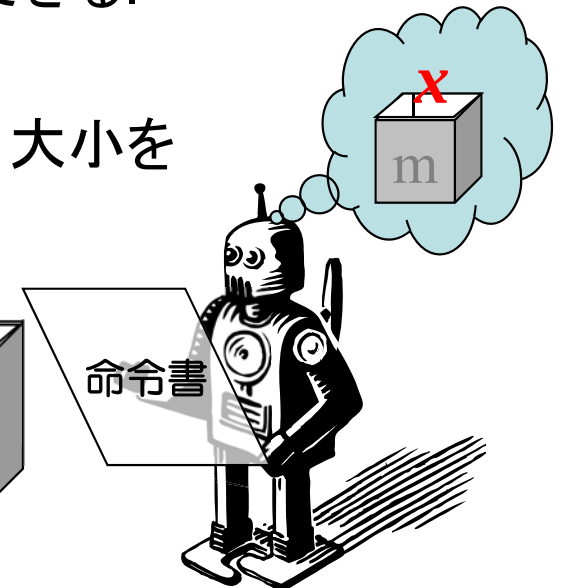
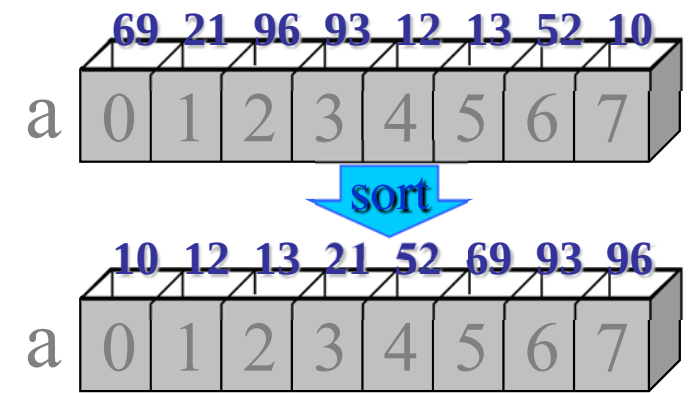
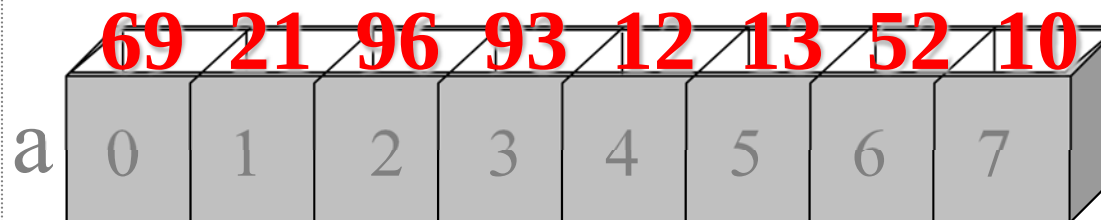
• 並び替え

– 下記ルールのもと、ロボットに昇順並替をさせる命令書を作れ。

- 8つの箱に8個の数字が入っている。
 - 箱には名前がついていて、箱の名前は, $a[0], a[1], \dots, a[7]$.
- ロボットは数字を一つだけ記憶できる。
 - ロボットの記憶用箱の名前は m . 新しい数字を記憶する場合は、前の数字に上書きされるが、一度覚えれば上書きされない限り記憶できる。
- ロボットは箱の中の数字を「消去」「記憶」「上書」できる。
 - ただし、一度に見ることができる箱は一つのみ。
- ロボットは、今見ている数字と記憶箱 m の数字の大小を「比較」できる。

命令の例

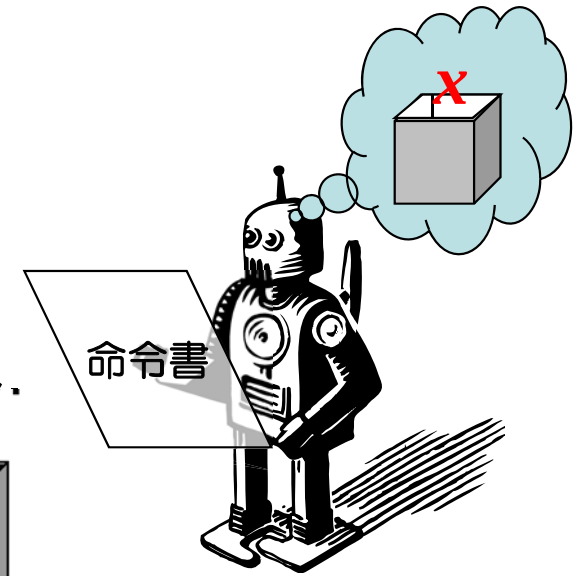
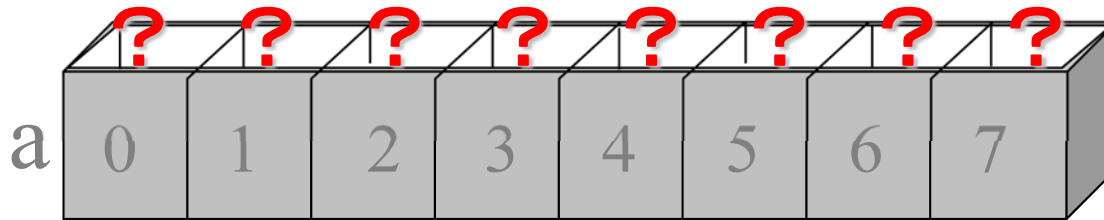
- 消去 $a[i]$
- 記憶 $a[i]$
- 上書 $a[i]$ by m
- 比較 $a[i]$ with m



アルゴリズム

- 並び替え

- ロボットに昇順並替をさせる命令書を作れ.



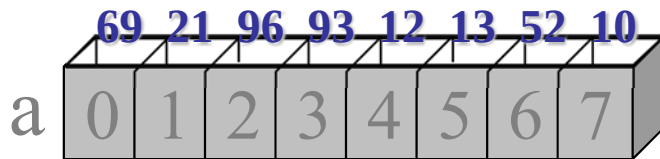
- 8つの箱に入っている8個の数字の値を, 我々は知らない(通常はこの通り, 事前にわからないことが殆ど). 前ページで, 数値がわかっていることを前提としたアルゴリズムだった場合は, 書き直せ.

- アルゴリズムの計算量

- あなたの考えたアルゴリズムは, 並び替えを終了するまでに何回(何ステップ)かかるか?
- 「消去」「記憶」「上書」「比較」をそれぞれ1回と数えて, これらが全部で何回必要か数えよ.
- 最も少ない回数のアルゴリズムを書いたのは誰かな?

アルゴリズム

- 並び替え



アルゴリズム記述例

【 $a[0], \dots, a[n-1]$ という n 個の数のソート】

Step1: $k := 0$ とする

Step2: $i := k+1$ とする.

Step2-1: $i=n$ なら Step3へ

Step2-2: もし $a[k] > a[i]$ なら, $a[k]$ と $a[i]$ を交換

Step2-3: $i := i+1$ として Step2-1へ

Step3: $k := k+1$ とする.

$k=n-1$ なら終了, そうでなければStep2へ

$k=0$ $i=1 \sim 7$ ($i=8$ で次のステップへ)

69, 21, 96, 93, 12, 13, 52, 10

21, 69, 96, 93, 12, 13, 52, 10

12, 69, 96, 93, 21, 13, 52, 10

10, 69, 96, 93, 21, 13, 52, 12

$k=1$ $i=2 \sim 7$ ($i=8$ で次のステップへ)

10, 69, 96, 93, 21, 13, 52, 12

10, 21, 96, 93, 69, 13, 52, 12

10, 13, 96, 93, 69, 21, 52, 12

10, 12, 96, 93, 69, 21, 52, 13

$k=2$ $i=3 \sim 7$ ($i=8$ で次のステップへ)

10, 12, 96, 93, 69, 21, 52, 13

10, 12, 69, 93, 96, 21, 52, 13

アルゴリズム

- 並び替えのアルゴリズム(基本編)

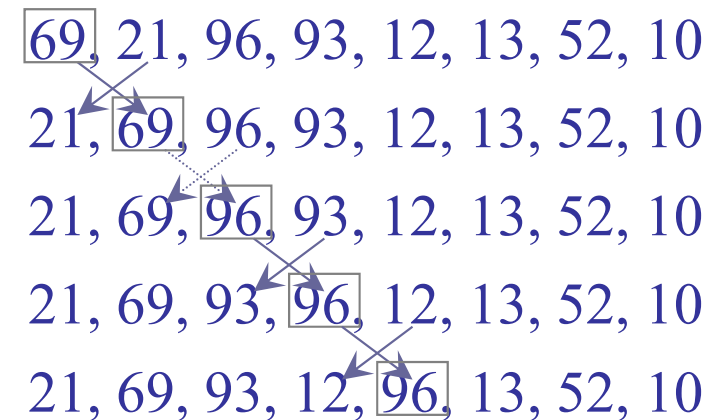
- 基本選択法(直接選択法)

- $a[i] \sim a[n-1]$ で最小項を探し $a[i]$ と交換,
を $i=0, \dots, n-2$ 繰り返し



- 基本交換法(バブルソート)

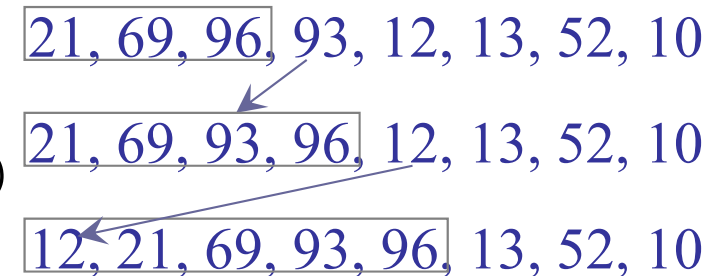
- $a[i] \sim a[n-1]$ で隣接2項を比較し交換(小左へ)
しつつ右へ移動(この操作で最大が最右へ),
を $i=0, \dots, n-2$ 繰り返し
- シェーカーソート〔バブルソートの改良版〕



- 基本挿入法

- 既整列部分列 $a[0] \sim a[i-1]$ に $a[i]$ を挿入
($a[i-1]$ と $a[i]$ を比較, $a[i-2]$ と $a[i]$ を比較, ...)
を $i=1, \dots, n-1$ 繰り返し

既整列部分列

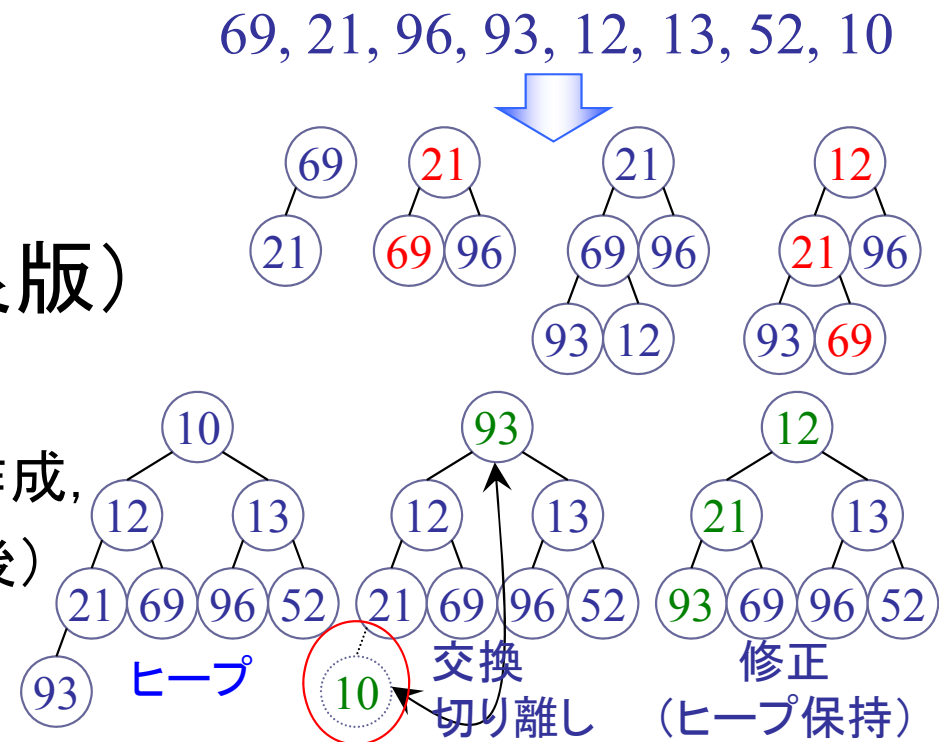


アルゴリズム

- 並び替えのアルゴリズム(改良版)

- 改良選択法(ヒープソート)

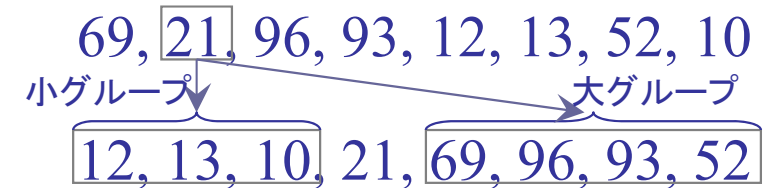
- ヒープ(親<子を満たす完全2分木)作成,
交換(ルートと最後)・切り離し(最後)
修正(ヒープの保持)の繰り返し



- 改良交換法(クイックソート)

- 数列中の適当な値を選び, その値
より小さい数を左, 大きい数を右に,
を繰り返し

10, 12, 13, 21, 52, 69, 93, 96



- 改良挿入法(シェルソート)

- gap毎に基本挿入法を行い,
gap縮小(例は1/2毎)を繰り返し

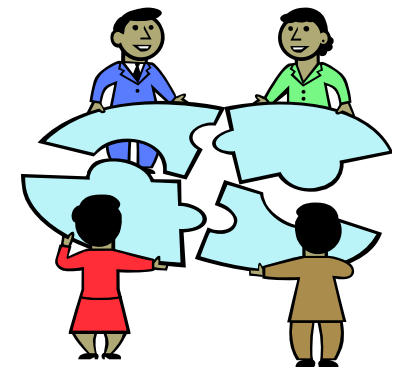
gap4(=8/2) 69, 21, 96, 93, 12, 13, 52, 10

gap2(=4/2) 12, 13, 52, 10, 69, 21, 96, 93

gap1(=2/2) 12, 10, 52, 13, 69, 21, 96, 93

- おおざっぱに → 詳細に

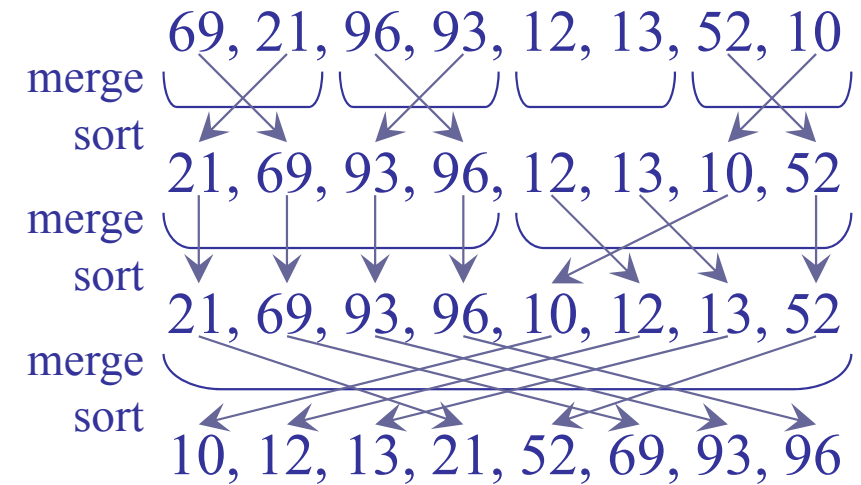
アルゴリズム



• 並び替えのアルゴリズム(その他のソート)

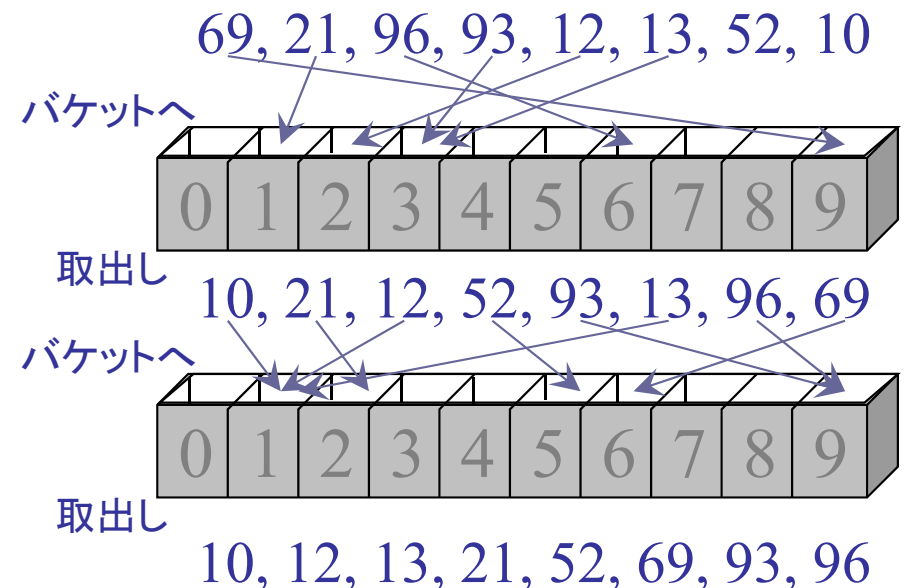
➤ マージソート

- 数列を1つずつの数値から結合(マージ)しつつ, ソートする
(結合の際, 小さい数同士を比較)

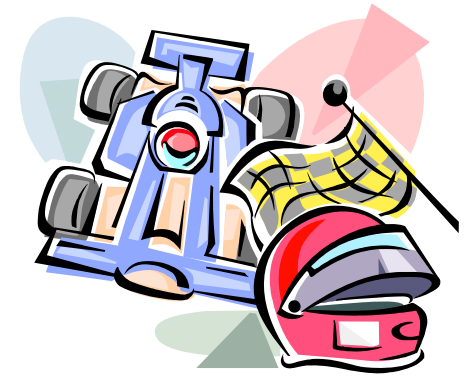


➤ ラディックスソート(基数ソート)

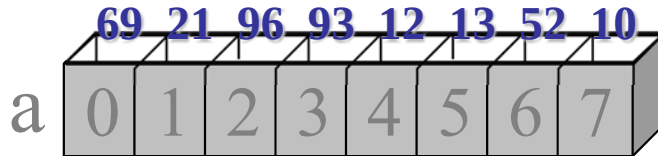
- 0~9のバケットを用意し, 数列の各数値の1の位を見て, 対応するバケットに入れる
- 0~9のバケット順に数値を取出し 並べる(各バケットは「キュー」)
- 10の位以降, 桁を上げて繰り返し
- 特徴は, 数値の「比較」をしないこと



アルゴリズム



- 並び替えの計算量



– 演習2: それぞれのアルゴリズムは実行にどのぐらいの時間が掛かりますか？

- 反復に要する回数は？
- 最悪計算量は？
- 最良計算量は？
- 平均計算量は？

例では

k=0: i=1,2,...,8

k=1: i=2,3,...,8

k=2: i=3,4,...,8

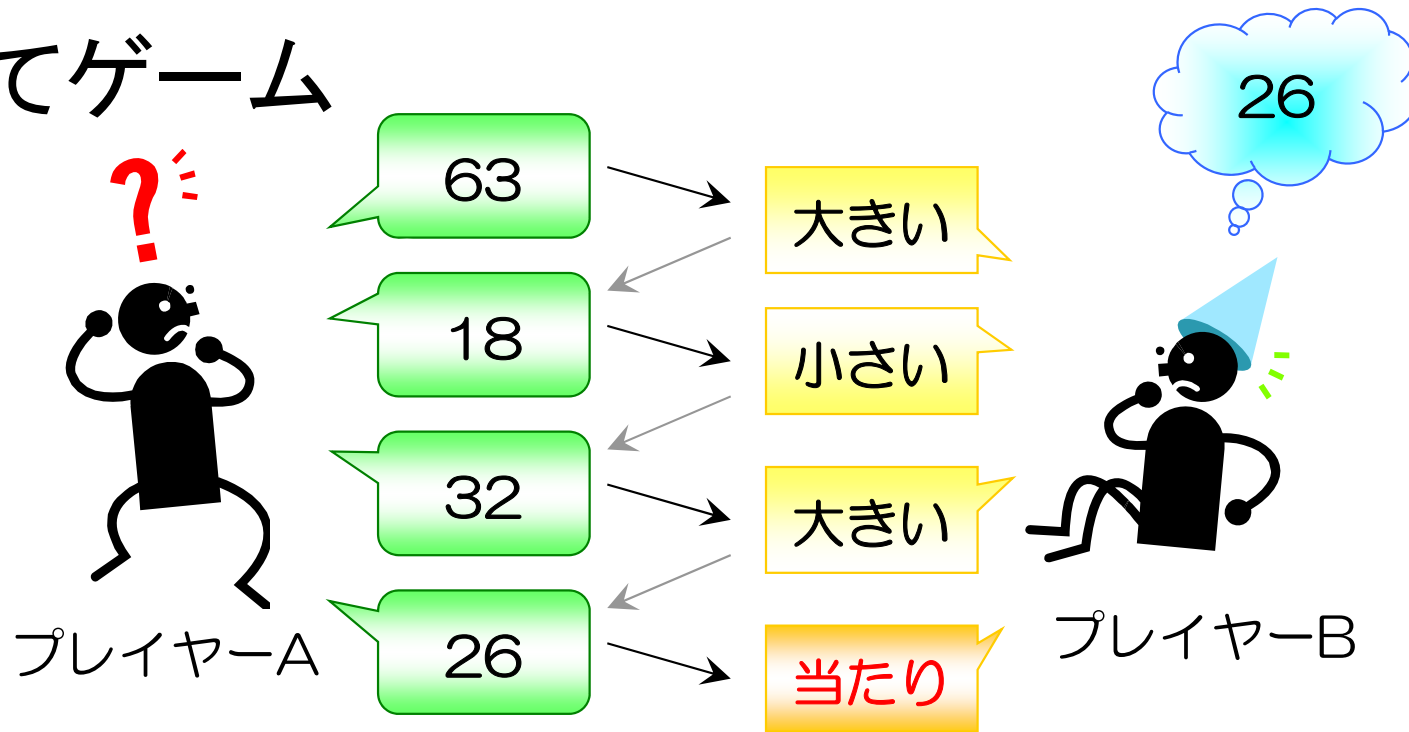
...

k=7: i=8 → 36回, $n(n+1)/2 = O(n^2)$

– 演習3: それぞれのアルゴリズムはどのぐらいのメモリ容量が必要ですか？

アルゴリズム

• 数当てゲーム

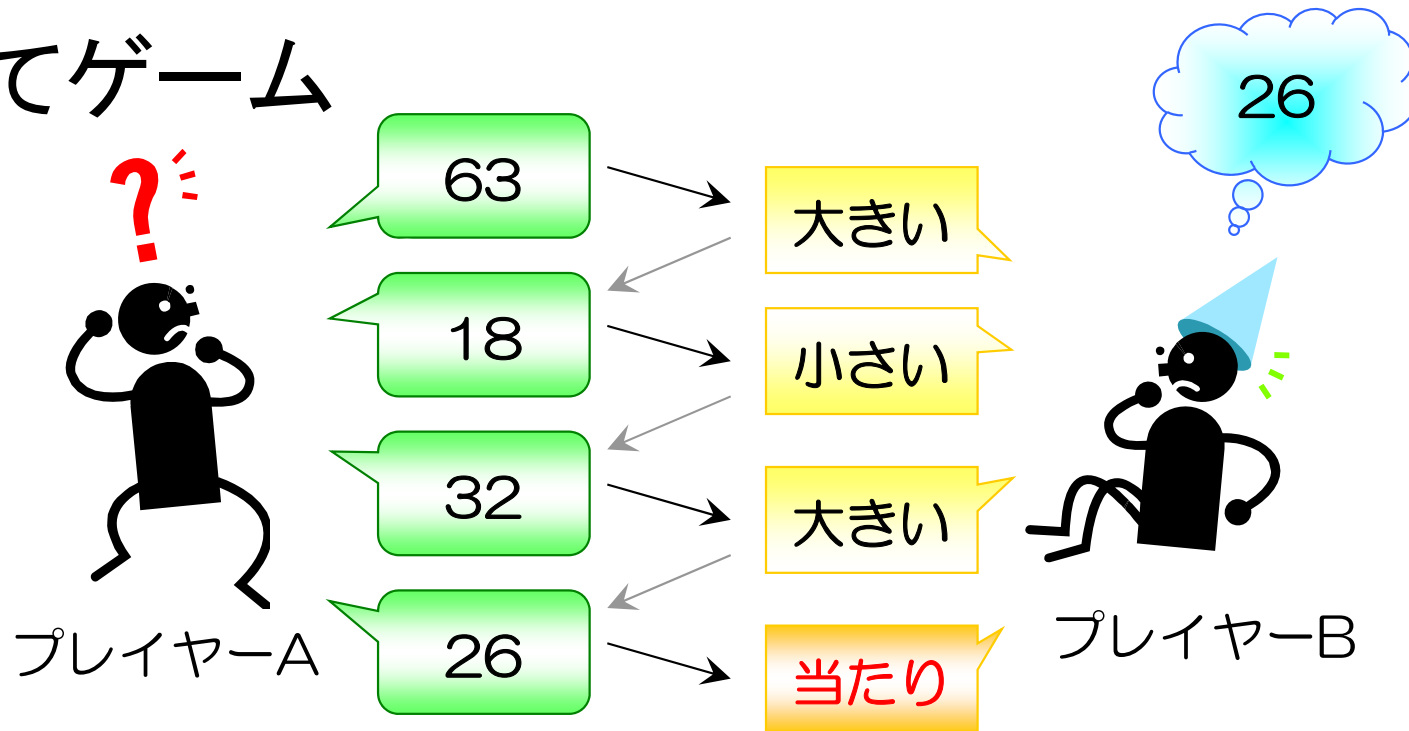


- 手順①: Bは1～100までの中の好きな数を一つ思い浮かべる →用紙に記入
手順②: Aはその数を当てる(数字を1つ言う) →用紙に記入
手順③: BはAが言った数に対し「当たり」「大きい」「小さい」を言う
手順④: ③の結果が「当たり」なら終了, そうでなければ②③を繰り返す

プレイヤーAとBの役割を交互に3回ずつ行い, 3回の平均点で争おう.
もちろん, 当てる回数が少ないほうが勝ちだ!

アルゴリズム

• 数当てゲーム



- プレイヤーAの役割の人は、何回言えば当たる？
- プレイヤーAのアルゴリズムを考えよう.
- プレイヤーBの役割の人は、どんな数字が当てられにくい？
- プレイヤーBの戦略を考えよう.

➡ コンピュータで対戦！ 誰のアルゴリズムが勝つかな？

アルゴリズム

- 数当てゲーム

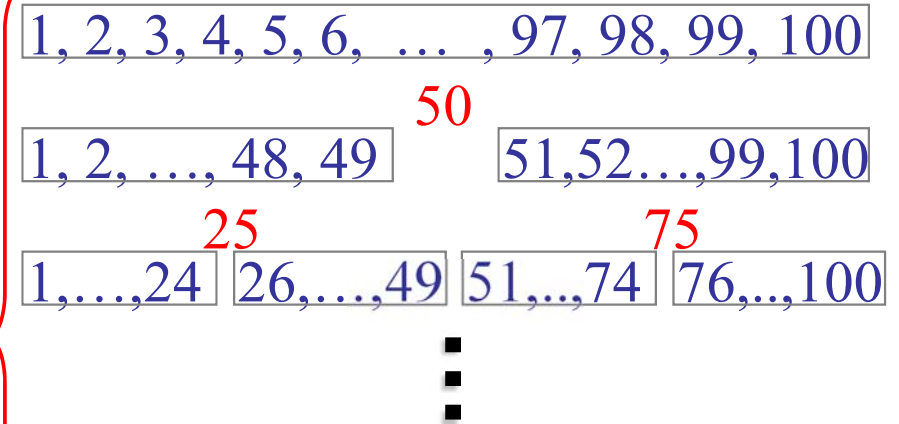
- 2分探索

- 探索範囲を半分(2分割)に減らすことを繰り返し

- $a[0], \dots, a[n-1]$

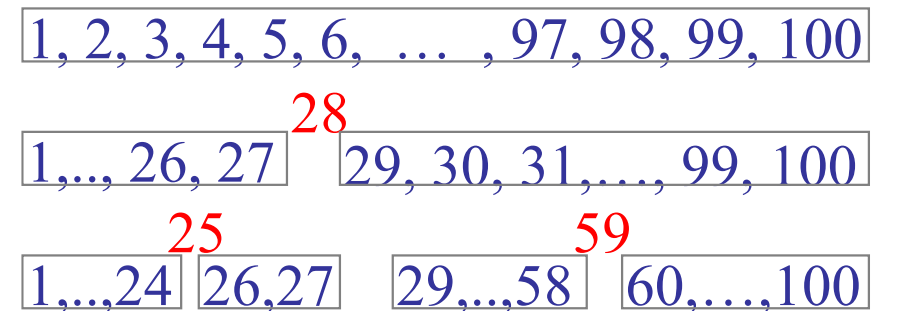
- $i := 0, j := n-1 \rightarrow \left\lfloor \frac{a[i] + a[j]}{2} \right\rfloor$

$\log_2 n$ 回の分割



- クイック(?)探索

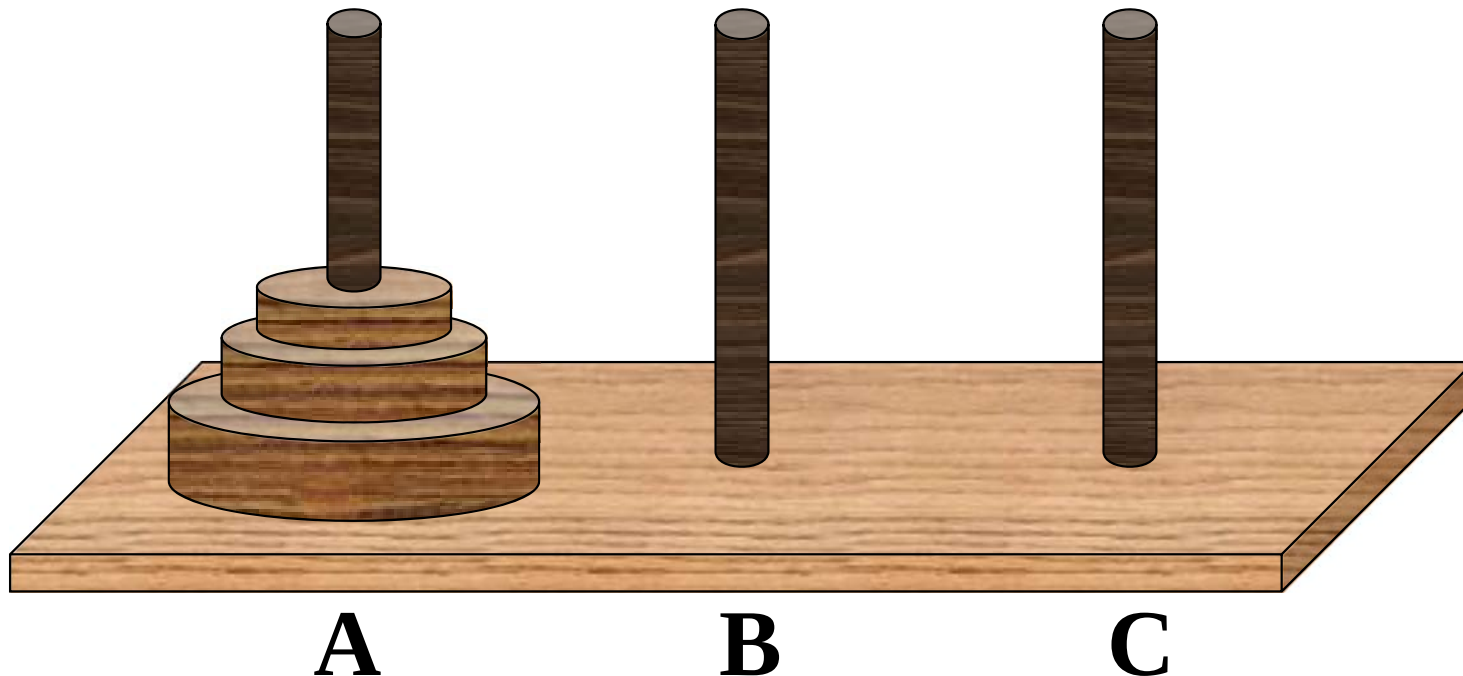
- 探索範囲を2分割するが、
分割基準はランダムに選ぶ



アルゴリズム

- ハノイの塔

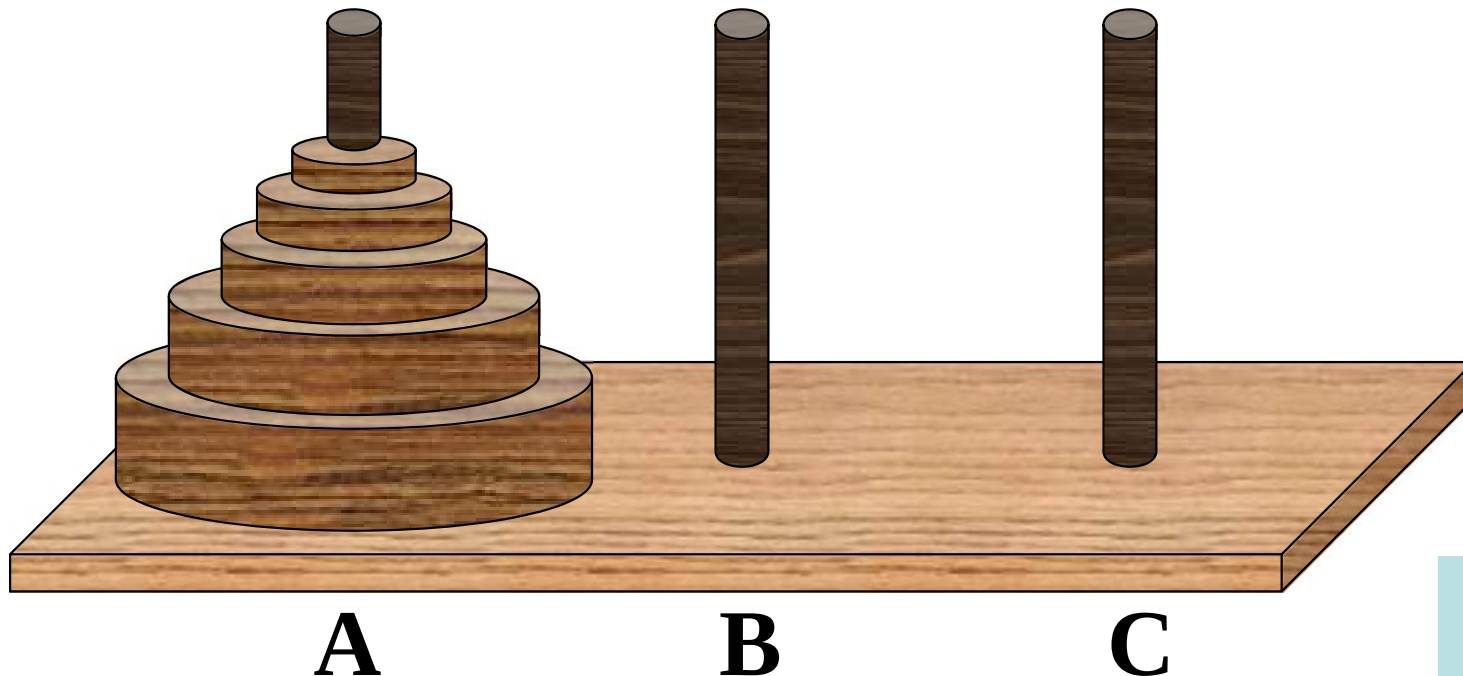
- Aにある3つの円盤を, 全てCに移したい
- ただし, 以下のルールを守らねばならない
 - 円盤は, 一度に1枚のみ移動できる
 - 小さな円盤の上に, 大きな円盤を置くことはできない
- 最小回数で円盤を移動させるアルゴリズムを書け



アルゴリズム

- ハノイの塔

- 円盤が4枚あったらどうなるか？
- 円盤が5枚あったらどうなるか？
- 宿題1: 円盤がn枚あったらどうなるか？ ($n=3,4,5,\dots,100,\dots$)
- 宿題2: 円盤が100枚あったら何秒かかるか？ (1回=1秒として)



アルゴリズム

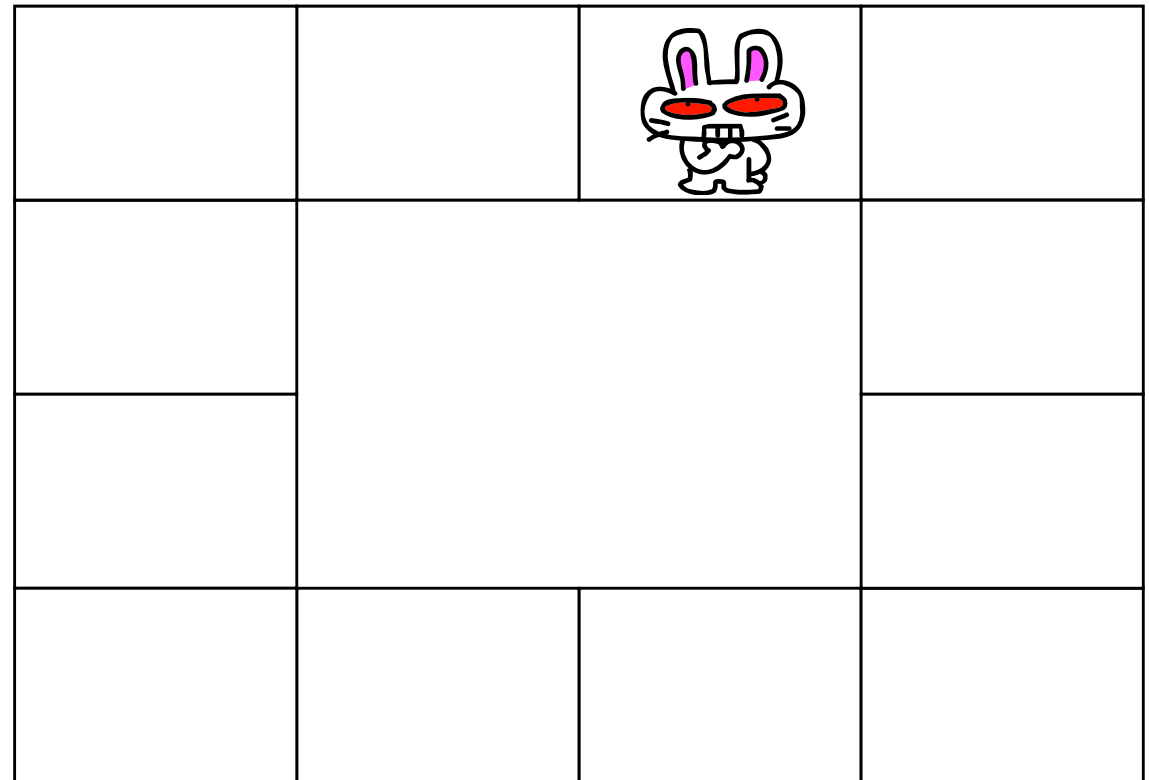
• ウサギと猟師

- 周回する狩場で、ウサギ・猟師は毎時刻どこか1マスに存在
- ウサギは最初に確率 $1/2$ で右か左に行くことを決め、以降はその方向に毎時間1マスずつ進む
- 猟師はウサギと同じマス目に入れば、ウサギを狩れる

- ウサギを捕まえるための
の猟師のアルゴリズムを
考えよ



- 捕まらないようウサギ
のアルゴリズムを考えよ



参考文献

- 浅野孝夫「情報の構造・上」日本評論社(1994)
 - ソート, アルゴリズム全般, 計算量について
- 浅野孝夫・今井浩「計算とアルゴリズム」オーム社(2000)
 - ソート, ラディックスソート, 計算量, ハノイの塔などについて
- 杉原厚吉「データ構造とアルゴリズム」共立出版(2001)
 - ソートについて
- 茨木俊秀「Cによるアルゴリズムとデータ構造」昭晃堂(1999)
 - ソートなどのアルゴリズムのCへの効率的実装について
- 河西朝雄「C言語によるはじめてのアルゴリズム入門」技術評論社(1992)
 - ソートの分類, ソート・ハノイの塔などのCプログラムについて