5.計算できないということ

計算できないとは

- 計算方法がわからない(定式化されていない)
- 計算にとても時間がかかる
- 計算するプログラムが存在しない

計算量

- 多項式時間計算量
  - 入力のサイズnに対して時間計算量がnの多項式(オーダが$O(n^k)$)となる時間計算量
  - 現実的な時間で計算可能
- 指数時間計算量
  - 入力のサイズnに対して時間計算量がnの指数関数(オーダが$O(k^n)$)となる時間計算量
  - 現実的な時間で計算不可能

問題と計算量

- ある問題を解くどのアルゴリズムでも、ある計算量より小さいオーダのアルゴリズムを作ることができない→計算量の下界(かかい)
  - n個の数の最大値を求める→n-1回の比較が必要
- 計算量の下界が非常に大きな問題も
  - プレスバーガー算術の判定(変数,0,1,+を組み合わせて作られる数式の等式から作られる条件が成り立つか否かを判定する問題)
  - 条件の長さnの時、最悪$2^{2^{cn}}$の計算量

問題の複雑さを考える

- NP完全問題の例(同じくらい複雑な問題)
- 部分和問題：正整数の組(b,a_1,…,a_n)が与えられた時、$b=\sum_{i\in S} a_i$ を満たす添字の部分集合Sは存在するか(ナップザック問題)
- 分割問題：n個の正整数(b,a_1,…,a_n)を分割し、各総和を等しくする $\sum_{i\in A} a_i = \sum_{i\notin A} a_i$となる部分集合 $A \subseteq \{1,\ldots,m\}(m \le n)$はあるか

NP完全問題の例

- 箱詰め問題：各々の容量が1であるような箱k個に、サイズが各々 $s_1, s_2, \ldots, s_n$ で $0 < s_i \leq 1$ の品物を箱詰めできるか
- 命題論理式の充足可能性問題：命題論理式の変数の値をどう割り当てると全体が真になるか

P∨(¬P)　Pが真でも偽でも全体は真(トートロジー)

P　　Pが真なら全体は真(充足可能)

P∧(¬P)　Pが真でも偽でも全体は偽(充足不能)

決定的･非決定的

- 決定的アルゴリズム
  - 計算の途中で複数の選択肢があるときに、ひとつずつ調べる
- 非決定的アルゴリズム
  - 計算の途中で複数の選択肢があるときに、全部を同時に調べる

クラスP・クラスNP

- 多項式時間限定:最悪でも時間計算量のオーダが多項式であるようなアルゴリズム

- クラスP : 多項式時間限定である問題の集合
- クラスNP : 多項式時間限定である非決定的アルゴリズムが存在するような問題の集合
  - 場合分けするとき、そのひとつひとつは多項式時間限定となるとき

P≠NPか

- クラスPよりクラスNPの問題の方が解くのに時間がかかりそう
- 本当にそうなのかはまだ未解決(P=NP問題と呼ばれる)

多項式時間還元可能

- 問題Π_1からΠ_2に多項式時間還元可能 $\Pi_1 \leq_m^p \Pi_2$ とは、関数Tが存在して、
  - 関数Tは多項式限定で計算できる
  - Π_1の各入力xについて、T(x)に対するΠ_2の答は、xに対するΠ_1の答と同じである
- ある問題から別の問題に変換可能であること
  - 二つの問題の難しさは同程度とみなす

問題 $\Pi_1 \xrightarrow{T} \Pi_2$

入力 $x \xrightarrow{T} T(x)$

答 a $\quad$ a

NP完全・NP困難

- 問題ΠがNP完全であるとは
  - ΠがNPに属する
  - NPに属する他の全ての問題Π'に対して $\Pi' \leq_m^p \Pi$
- 問題ΠがNP困難であるとは
  - NPに属する他の全ての問題Π'に対して $\Pi' \leq_m^p \Pi$

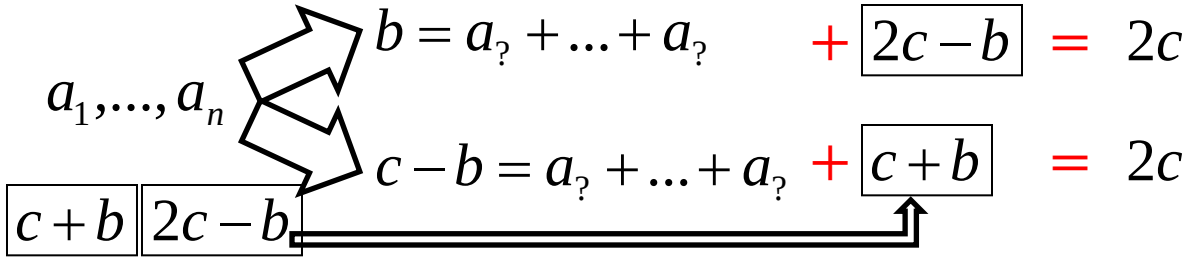
問題の変換の例

- 「部分和問題は、分割問題に多項式時間還元可能である」
  - 部分和問題の入力 $(b, a_1, \ldots, a_n)$ $\quad c = a_1 + \ldots + a_n$
  - $a_1, \ldots, a_n$ の一部で和がbなら、残りはc-b

  →入力として $c+b$ と $2c-b$ を追加→ $a_1, \ldots, a_n, c+b, 2c-b$

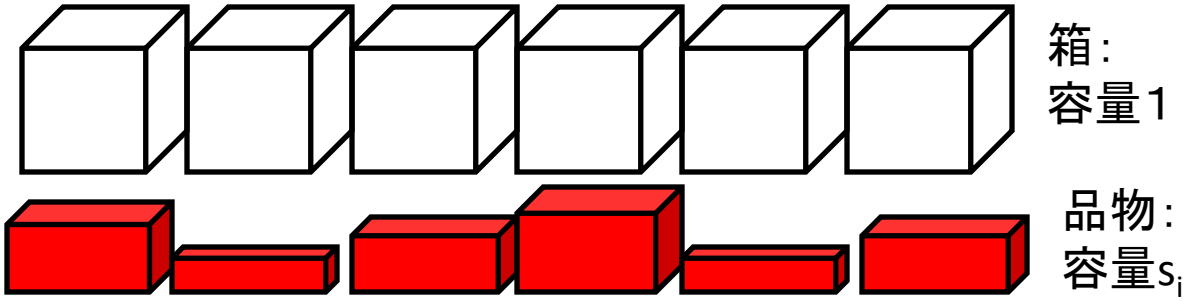
  和がbとc-bになるよう $a_1, \ldots, a_n$ を分けることは

  和が2cと2cになるよう $a_1, \ldots, a_n, c+b, 2c-b$ を分けること



近似アルゴリズム

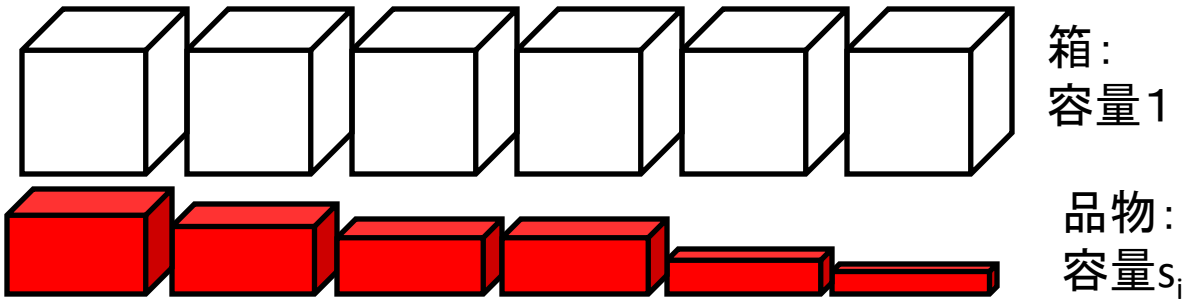
- NP完全問題を解くには指数時間計算量
  →それに近い答を多項式時間計算量で解きたい
- 例：箱詰め問題
  – 全ての品物を詰めるのに箱がいくつ必要か



ファーストフィット法

- 品物を大きい順に並べ替える
- 最初のものから順番に箱に詰めていく
- 箱があふれたら、次の箱を使う
- 全部詰め終わったら終わり
- (ファーストフィット法での箱の数)

$$\leqq \frac{11}{9} \times (\text{最適な箱の数}) + \frac{6}{9}$$



計算可能性

- 計算とは: 入力xに対し、関数f(x)を出力する(問題が定式化されている)
- 計算可能とは: 入力xに対し関数f(x)を出力するプログラムが存在
- 計算不可能とは: 入力xに対し関数f(x)を出力するプログラムが存在しない

チューリング機械上で

- チューリング機械とは…
  - 2年前期の「計算基礎論」で勉強します

計算可能でない問題の例

- 「不定方程式(整数係数の連立n次方程式)の整数解は存在するか否かを有限回の手段で判定する」
  - 2変数の場合は可能
  - 一般の場合は不可能

計算不可能な関数

- 自然数から自然数への関数f:N→Nで、計算可能でないものが存在(N:自然数全体の集合)
- 証明:背理法
  - 関数fを全て並べる $f_0, f_1, f_2, f_3, \ldots$
  - 新たな関数$f(i)=f_i(i)+1$を定義→fはその中のいずれか→N番目の関数であるとする→ $f_N(N)=f_N(N)+1$ となり矛盾

プログラムの停止問題

- 「プログラムPが与えられた時、どんなデータが入力として与えられても停止するか、そうでないかを判定する」のは計算不可能
- 証明：背理法

プログラム　データ

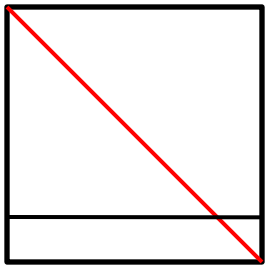
  - 判定できるなら判定するプログラムHを作る。H(P,D)は停止するならyes,それ以外はno
  - ひねくれたプログラム $G(P) = \begin{cases} H(P,P) \text{ がyesなら停止しない} \\ H(P,P) \text{ がnoなら停止} \end{cases}$
  - Gの入力にG(プログラム自身)

$$G(G) = \begin{cases} H(G,G) \text{ がyesなら停止しない} \\ H(G,G) \text{ がnoなら停止} \end{cases}$$

$H(G,G)$ はyes? no?→矛盾

対角線論法

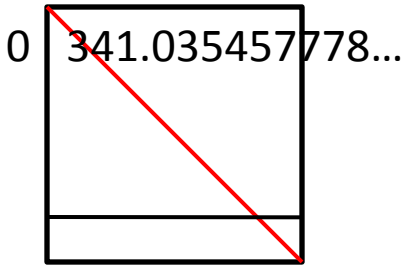
- 対角線は、どの行(列)ともいつかは交わる



- ある集合の要素を全て並べる
- 対角線上の値(赤)を変換したものを考える
- この赤線は、どの行(列)とも一致しない→全て並べたことと矛盾

カントールの定理

- N(自然数全体の集合)からR(実数全体の集合)への全単射は存在しない
- 全射:全てのyに対してy=f(x)を満たすxが存在
- 単射:$x_1 \neq x_2$なら$f(x_1) \neq f(x_2)$
- 証明:背理法
  - 全ての実数を並べた表
    →行数とその行の実数が対応
  - 対角線上の数値で0なら1、それ以外なら0となる数を作る(実数r)
  - この実数rはどの行とも一致しない←全ての実数を並べたことに矛盾



計算不可能な関数

- 自然数から自然数への関数f:N→Nで、計算可能でないものが存在(N:自然数全体の集合)
- 証明:背理法
  - 関数fを全て並べる $f_0,f_1,f_2,f_3,\ldots$
  - 新たな関数$f(i)=f_i(i)+1$を定義→fはその中のいずれか→N番目の関数であるとする→$f_N(N)=f_N(N)+1$となり矛盾

これが対角線

$f_i(i)$

関数の番号

データ

関数番号 \ データ	0	1	2	3	…
0	$f_0(0)$	$f_0(1)$	$f_0(2)$	$f_0(3)$	
1	$f_1(0)$	$f_1(1)$	$f_1(2)$	$f_1(3)$	
2	$f_2(0)$	$f_2(1)$	$f_2(2)$	$f_2(3)$	
3	$f_3(0)$	$f_3(1)$	$f_3(2)$	$f_3(3)$	
…					

プログラムの停止問題

データ

プログラム

	0	1	2	…
0	○	×	×	…
1	×	×	○	…
2	○	×	○	…
…	…	…	…	…

「×○×…」というプログラムはどの行とも一致しない

プログラム　データ

- 証明：背理法
  - 判定できるなら判定するプログラムHを作る。H(P,D)は停止するならyes,それ以外はno
  - ひねくれたプログラム $G(P)=\begin{cases} H(P,P) & \text{がyesなら停止しない} \\ H(P,P) & \text{がnoなら停止} \end{cases}$
  - Gの入力にG(プログラム自身)

$$G(G)=\begin{cases} H(G,G) & \text{がyesなら停止しない} \\ H(G,G) & \text{がnoなら停止} \end{cases}$$

$H(G,G)$ はyes? no?→矛盾