授業構成と成績評価

- 1講時は講義，2講時は演習
- 講義資料の配布や、授業関係の連絡はすべてTeamsにて行う
- 講義資料は約1週間前に配布する（予習できるように）
- 演習課題は講義資料に含まれる
- 演習課題の提出：課題を完成した時点でTAを呼び、確認してもらう
- 演習課題の提出期限：その都度知らせる
- レポート課題がある
- 「確認テスト」を第8回（最終回）の授業で実施する
- 成績評価：演習50%, レポート15%, 確認テスト35%

新型コロナウイルス感染者･濃厚接触者について

- 保健管理センター（感染等報告窓口）075-645-1168に連絡すると同時に、教員にも速やかに連絡ください。
- 教員への連絡はTeamsのチャットを利用ください。
- 該当者へは、以下の配慮を施す
  1. 欠席期間中の授業欠席を理由ありの欠席として認める
  2. 欠席期間中の演習課題の提出を不要とする
- なお、本授業は教室対面実施のみなので、オンライン参加はできない。また、録画資料も利用できない。講義資料で自学し、わからない箇所などがあれば授業参加時にまたはTeamsのチャットで教員に質問してください

演習

- 教室のパソコンのLinux(Ubuntu)または携帯PCを利用する
  - 教室PCと携帯PC間のデータのやり取りはたとえばGoogle Driveを通じて行うことができる
- 質問等があれば挙手、TAさん・教員が直接対応
- 課題のチェックも挙手、TA・教員が直接対応

導入

アルゴリズムとは

まず、クイズを

- 1±2±3±4±5±6±7±8±9±10=11は何通り？

  ✓例：1+2-3-4+5+6-7-8+9+10=11

  ✓可能の場合数：±が9個→2の9乗→512通り

  ✓この512通りの場合から漏れなく，正しく，効率よく11となる場合をすべて選び出すにはどうすればよいか

  ✓このような求め方（解法）を考えるのがアルゴリズムの話

正解（18通り）

1+2+3+4+5-6-7+8-9+10
1+2+3+4-5+6+7-8-9+10
1+2+3+4-5+6-7+8+9-10
1+2+3-4+5+6+7-8+9-10
1+2+3-4-5-6-7+8+9+10
1+2-3+4+5+6+7+8-9-10
1+2-3+4-5-6+7-8+9+10
1+2-3-4+5+6-7-8+9+10
1+2-3-4+5-6+7+8-9+10

1+2-3-4-5+6+7+8+9-10
1-2+3+4-5+6-7-8+9+10
1-2+3+4-5-6+7+8-9+10
1-2+3-4+5+6-7+8-9+10
1-2+3-4+5-6+7+8+9-10
1-2-3+4+5+6+7-8-9+10
1-2-3+4+5+6-7+8+9-10
1-2-3-4+5-6-7+8+9+10
1-2-3-4-5+6+7-8+9+10

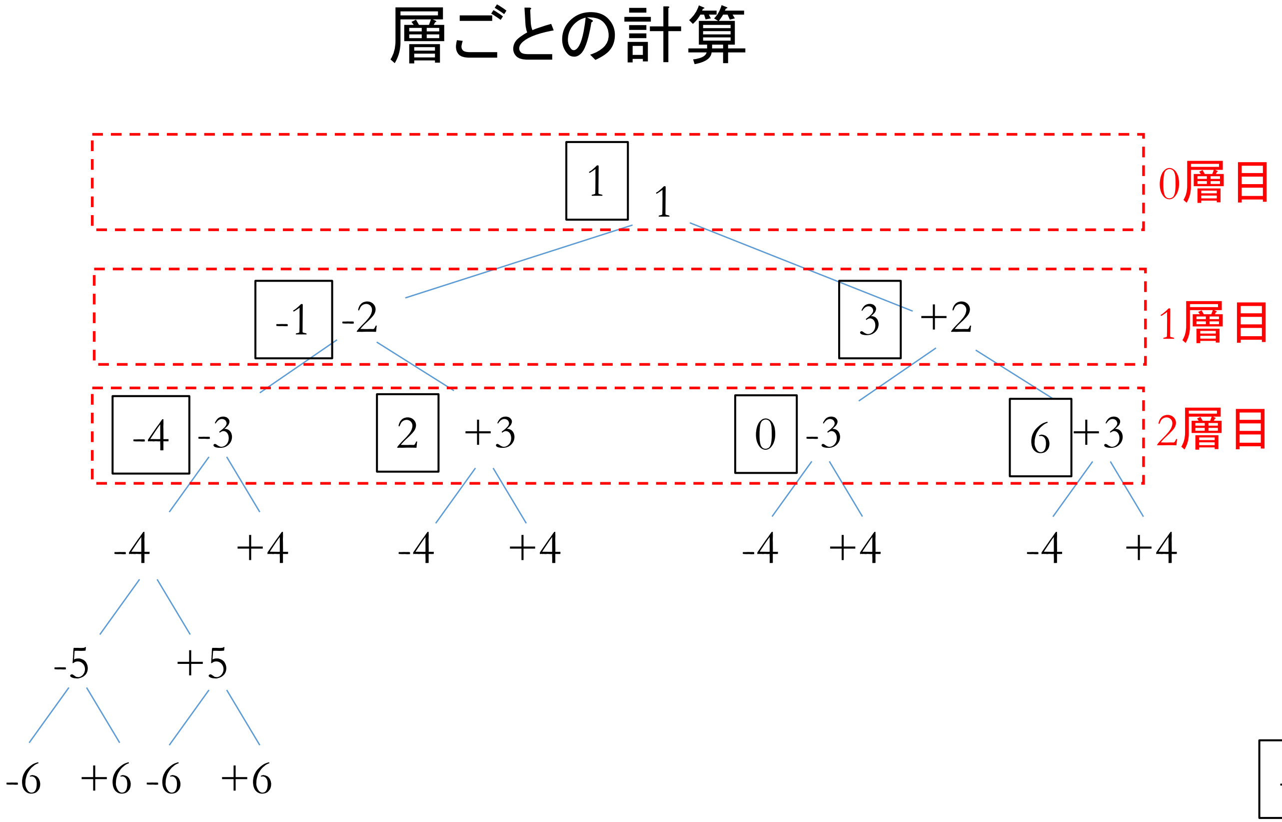
解き方いろいろ

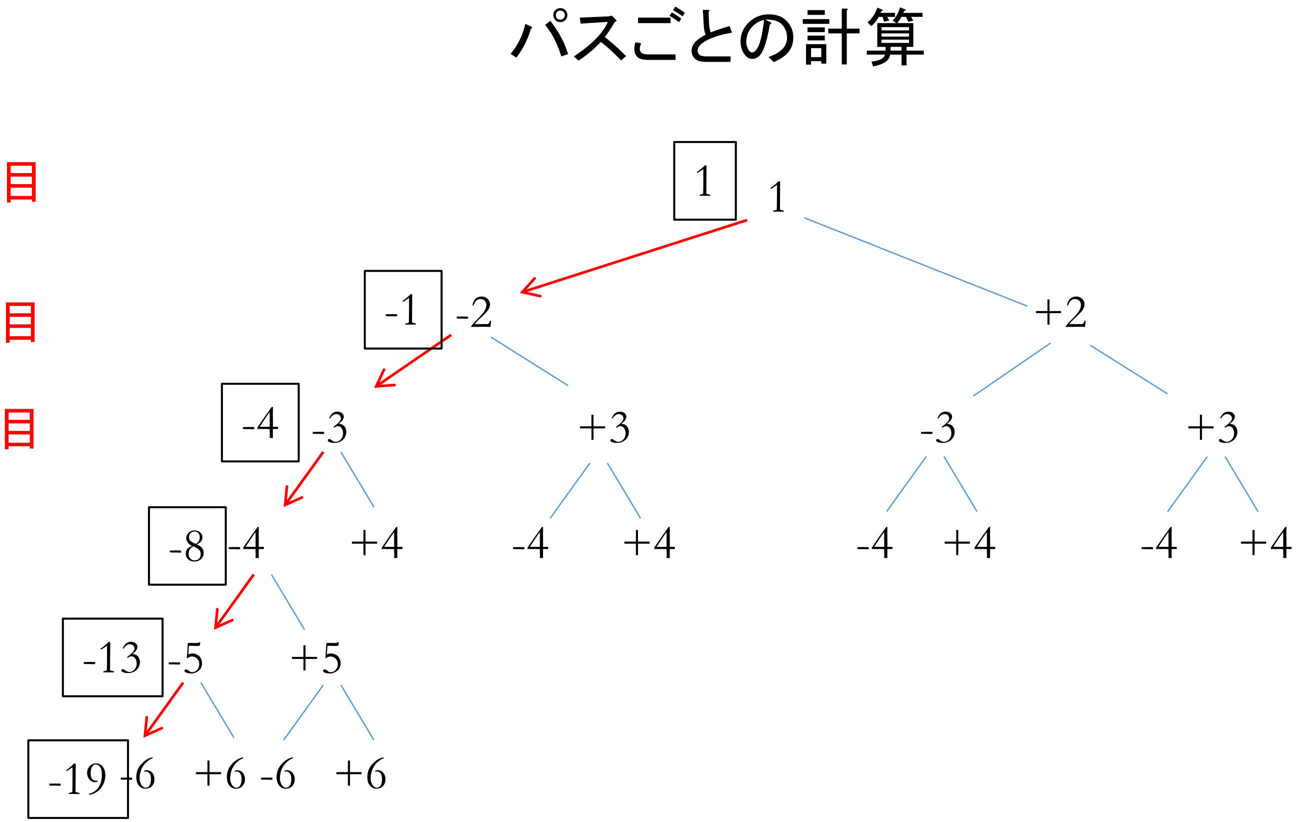
- 全数列挙法
  - 全512通りの計算を行う方法
- 枝刈り法
  - 計算の途中で可能性がなくなれば打ち切る
- 分割統治法
  - 問題を分割し、より小さい問題を解き、部分解をまとめて全体の解を得る
- 整数分解法
  - 全部足したら55なので、プラスとマイナスを分け、プラスの和が33またはマイナスの和が22になるような組み合わせを求めればよい

全数列挙法

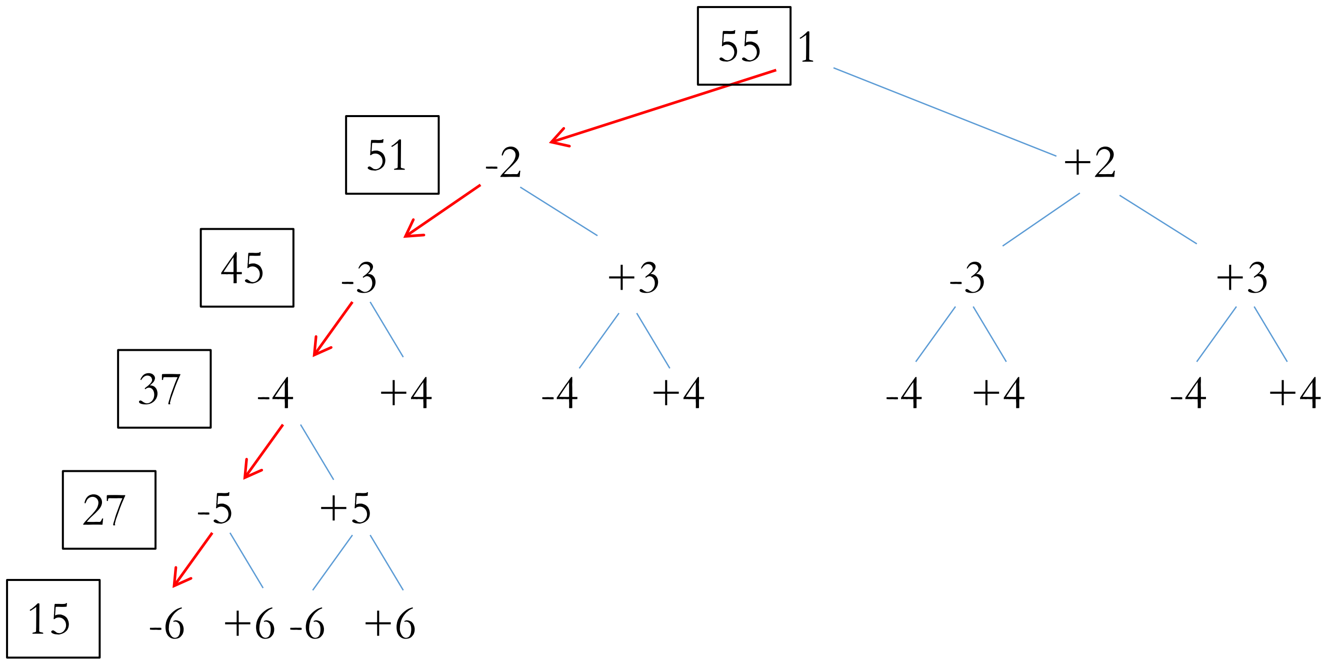
- ポイントは、1±2±3±4…の全512通りの組み合わせ(計算式)をどう作るか
- (2分木による)±を場合分けして行き、すべての場合を作ればよい
- 層ごとの計算
  - 2分木を根から各層のノードまでの合計値を計算していき、計512個の中から値が11のものを数える
- パスごとの計算
  - 深さ優先探索法を利用して、根から葉ノードまでの各パスの合計値を計算し、計512個の中から値が11のものを数える

全数列挙法


層ごとの計算
1
1
-1
-2
3
+2
-4
-3
2
+3
0
-3
6
+3
-4
+4
-4
+4
-4
+4
-4
+4
-5
+5
-6
+6
-6
+6
0層目
1層目
2層目


パスごとの計算
1
1
-1
-2
+2
-4
-3
+3
-3
+3
-8
-4
+4
-4
+4
-4
+4
-4
+4
-13
-5
+5
-19
-6
+6
-6
+6

枝刈り法



7を-7とおけば値11未満となるので、それ以上の計算（探索）は打ち切る。代わりに+7から計算（探索）を続ける

分割統治法

1組（8個）	2組（8個）	3組（8個）
1+2+3+4=10	5+6+7=18	8+9+10=27
1-2+3+4=6	-5+6+7=8	-8+9+10=11
1+2-3+4=4	5-6+7=6	8-9+10=9
1+2+3-4=2	5+6-7=4	8+9-10=7
1-2-3+4=0	-5-6+7=-4	-8-9+10=-7
1-2+3-4=-2	-5+6-7=-6	-8+9-10=-9
1+2-3-4=-4	5-6-7=-8	8-9-10=-11
1-2-3-4=-8	-5-6-7=-18	-8-9-10=-27

1組と2組と3組の値を足して11になるかを調べる

整数分解法

- すべて+で足したら55なので、正の数の合計が33or負の数の合計が-22の組み合わせを探せばよい。正の数に着目した場合、たとえば

  10, 9, 8, 5, 1(-7, -6, -4, -3, -2)

  10, 9, 8, 3, 2, 1(-7, -6, -5, -4)

  ⋮

Question

1. 1±2±3±4±5の組み合わせは何通りか
2. 1±2±3±4±5=5というクイズについて、数1～5を(1,2,3)と(4,5)に分割し、分割統治法で解をもとめなさい
3. 上記問題を整数分解法で解く場合、プラスの値の合計値がいくつでマイナスの値の合計値がいくつであれば、このような組み合わせが解となるか。また、これらの組み合わせを求めなさい

クイズ問題の規模が大きくなると

- たとえば1±2±... ±20=21

⇩

- 計2^{19}=524,288通りの組み合わせの計算が必要

⇩

- 人手では手に負えなくなる

⇩

- 計算機の力を借りるしかない（計算機で解くしかない）

⇩

- これまでの「解法」だけわかっていても、計算機にどう処理させればよいかがわからない

解法からアルゴリズムへ

アルゴリズムとは

- 解法が分かったとしても、計算機での実現までにはまだ距離がある
- 解法を、計算機で実現する手順で、表現しなければならない
- 手順は、どのような処理をどのような順序で行うかを、曖昧な点の残らないようにきちんと定めたものでなければならない
- このような手順をアルゴリズム（Algorithm）という
- しかしプログラムも手順といえば手順である。何が違うだろう？
  - アルゴリズムは、手順を人間向きの言葉で表現したものである（言語依存しない。設計にあたる）。計算機に理解させるためにはそれをプログラムに変換する必要がある
  - 言い換えれば、アルゴリズムは解法とプログラム間の中間表現であり、プログラムはその解法の計算機上の（真の）実現である

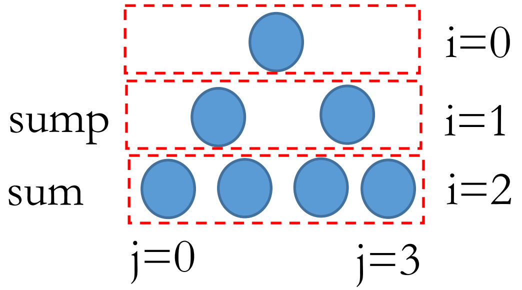
例：層ごと計算のアルゴリズム

入力：データの種類数n=10とデータの種類の配列a[]={1,2,3,4,5,6,7,8,9,10}

出力：答えが11となる組み合わせの数

補助: 層内の各ノードの合計値配列：前の層用sump, 現在の層用sum,

層内のノード数n_node, 層番号i, （層ごとに用いる）ノード番号j, k



1. {初期化}n_node=1, i=j=0, sump[j]=a[j]とおく
2. {各層について}i=1からn-1まで以下の処理を繰り返す

   2.1 k=0とおく

   2.2 {層内のノードについて}j=0からj=n_node-1まで以下の処理を繰り返す

      2.2.1 sum[k]=sump[j]+a[i], k++

      2.2.2 sum[k]=sump[j]-a[i], k++

例：層ごと計算のアルゴリズム

2.3 n_node=k

2.4 j=0からj=n_node-1まで以下の処理を繰り返す

sump[j]=sum[j]

3. {最後の層の各ノードの合計値の答え合わせ}

3.1 k=0とおく

3.2 i=0からi=n_node-1まで以下の処理を繰り返す

もしsum[i]=11ならk++

以上でkが答えとなる

アルゴリズムの評価

アルゴリズムの計算量

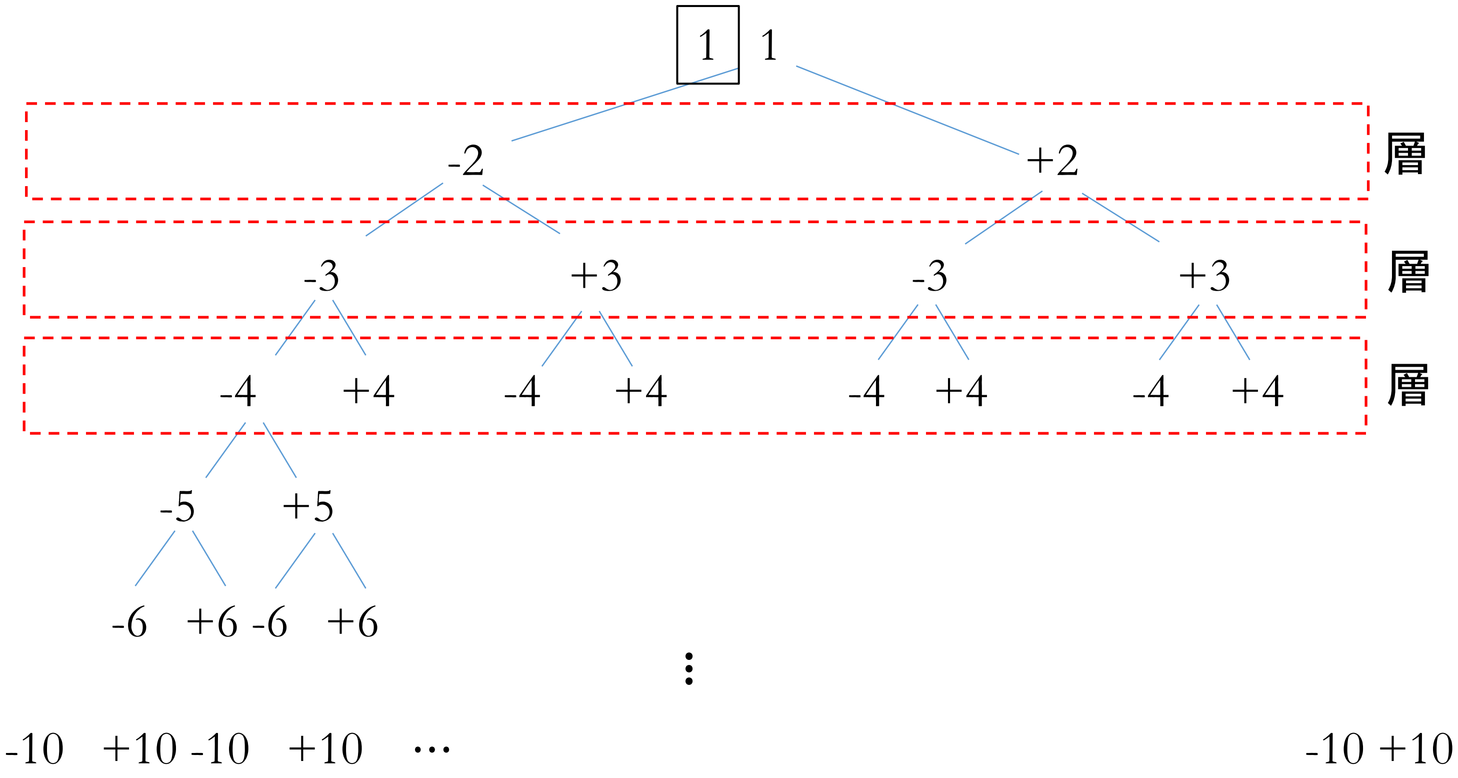
- 上述のように、クイズの解を漏れなく正しく得るアルゴリズムはいろいろと考えられる
- では、効率よく解けるのはどれだろうか？
- そもそも、効率って何？どう測ればよい？

- 計算量の話

アルゴリズムの計算量

- 計算機がそのアルゴリズムの実行に要する計算資源の量であり、時間計算量(処理時間)と空間計算量／領域計算量(必要なメモリ)がある
- 両方の計算量が重要だが、空間計算量の把握が困難な場合が多く、現在、メモリなどが安価なこと、また、処理時間が大事などを総合的に考え、時間計算量がより重要視される
- 本授業では以降、「計算量」のみを使う場合、時間計算量を指す
- 以下はクイズ(1±2±3±4±5±6±7±8±9±10=11の場合)の各解法の計算量を見てみよう。ここでは、足し算または引き算の回数を計算量とする

全数列挙法の計算量



$$2^1 + 2^2 + \cdots + 2^9 = \frac{2\left(1 - 2^9\right)}{1 - 2} = 2 \times 511 = 1022$$

枝刈り法の計算量

- 正確な計算は困難なので、直接プログラムで確認した
- その結果、計算量はわずか353であった
- これは、全数列挙法の1022の約1/3であった
- ちなみに葉ノードまで到達した経路（枝刈りされなかった経路）は169で、これは全512経路の約1/3であった

いろいろな計算量

- 時間計算量には、時間計算量、平均時間計算量、最悪（最大）時間計算量がある
- 時間計算量：データが固定の場合、またはデータの与え方に依存しない場合の計算量
  - 前述の計算量はデータ固定のため、これに相当する
- 平均時間計算量とは、データの与え方に依存する場合、平均的なデータが与えられたときの計算量を示すものである
  - 例えば、線形探索アルゴリズムにおいては、中央のデータが与えられたとき必要とされる比較回数が平均計算量になる
  - 求めるのは一般的に難しい

いろいろな計算量

- 最悪（最大）時間計算量とは、最も具合の悪いデータが与えられたときの計算量を示すものである
  - 実用上完璧さを要求する場合が一般的なので、よく使われる

平均計算量と最悪計算量の計算例

- 線形探索を例とする
- 線形探索とは一列に並んでいるデータ(たとえば、5,3,7,10,9,...)を先頭から探して(比較して)いく探索である
- ここで計算量をデータの比較回数とする
- 探索データが先頭にあれば比較が1回で済む。一方、最後にあれば比較はn回しなければならない

- データの数がnだとすると、線形探索の平均計算量と最悪計算量はそれぞれ以下となる

  平均:(1+n)/2

  最大:n

計算量の漸近的な評価

- 計算量は、上記例のように、問題の大きさを表すパラメータnを設定し、その関数として表すのが一般的である
- 計算量がnの関数で表されたら、次に計算量の漸近的な評価を行うことが重要
- 漸近的な評価って？漸近的な評価はなぜ重要？
- たとえば、同じ問題が解ける2つのアルゴリズムがあって、その計算量がそれぞれ$f_1(n)=1000n$, $f_2(n)=0.5n^3$だとする。さて、どちらのほうが効率がよいだろう

計算量の漸近的な評価

- 漸近的な評価とは、データの数nが非常に大きい場合の評価

n	$f_1(n)=1000n$	$f_2(n)=0.5n^3$
1	1000	0.5
100	100000	500000
10000	10000000	500000000000

- $f_2(n)$の漸近計算量>$f_1(n)$の漸近計算量
- なお、上記の表に示しているように、計算量の増加は定数（1000, 0.5など）よりは主にnによって決まる。しかも、1000nよりも$0.5n^3$の方がはるかに影響が大きい
- つまり、たとえば$f_3(n)=1000n+0.5n^3$を漸近的に評価するときは、次数が一番大きいn^3の項のみを考えればよい

オーダー

- 上記のような表をいちいち求めなくてもよいように、導入されたのがオーダーの概念
- 「データ量を大きくしていったときの計算量は次数の一番大きい項に大きな影響を受け、その他の要因にはあまり影響を受けない」ということから、その次数の一番大きい項を表現するために用いられる表現方法である

$f_3(n)=1000n+0.5n^3 \Rightarrow f_3(n)=O(n^3)$ （$f_3(n)$のオーダーはn^3である）

$f_1(n)=1000n \Rightarrow f_1(n)=O(n)$

$f_2(n)=0.5n^3 \Rightarrow f_2(n)=O(n^3)$

- $f_1(n)$の計算量が一番小さくて残りの2つが同じである

Question

- ある問題を解く3つのアルゴリズムA, B, Cの計算量は次のように求められたとする

  $f_A(n)=1000n^3$, $f_B(n)=10000n+100n^2\log n$, $f_C(n)=10n^4$

  どのアルゴリズムが一番性能がよいかをそれぞれのオーダーを計算した上で答えなさい

Question

- 同じ問題に対し、計算機Aに実装したプログラムは計算量nのアルゴリズムを用い、計算機Bに実装したプログラムは計算量n^3のアルゴリズムを採用した。さて、計算機Aの処理速度が1000倍速くなったとすれば、同じ時間内で解ける問題の規模は何倍になるか。また、計算機Bも1000倍速くなったとしたら、同じ時間内で解ける問題の規模は何倍になるか。

本授業は

- ソート・探索などの基本アルゴリズムや、Nクイーンクイズのような組み合わせ的な問題の解き方を習得するとともに、その習得過程を通じて、与えられた問題に対し、効率のよいアルゴリズムを創り出す能力を身に着けること、つまり、問題解決能力を養うことを目的としている

演習課題に取り組む前に

- まず、homeディレクトリ（Ubuntuであればログインして端末を立ち上げたところ、WindowsのWSL/WSL2であればWSLを立ち上げたところ）で、algoという名前のディレクトリ（フォルダ）を作成してください
  - mkdir algo
- 次に、このディレクトリの下に移動してください
  - cd algo
- 演習課題への取り組みはalgoというディレクトリ（フォルダ）の下で行ってください
  - code プログラム名.c　　プログラムの作成
  - cc プログラム名.c　　コンパイル
  - ./a.out　　実行

第1回演習課題

1. 全数列挙法の層ごと計算のアルゴリズムに少し修正を加え、$2 \leq 1 \pm 2 \pm 3 \pm 4 \leq 8$が成立する場合（組み合わせ）の数を求めるプログラム（ex01-comb.c）を作成しなさい。

第1回演習課題

2. （発展問題）a1±a2±a3±a4≥ vが成り立つ場合の数を求める問題を、a1, a2, a3, a4をa1, a2とa3, a4に分割して分割統治法で求めたい。このようなプログラム（ex01-comb-dc.c）をできるだけ簡潔に作成しなさい。ただし、a1, a2, a3, a4は正の整数、vは整数で、標準入力で与えるとする。

   ヒント：int sign[]={-1, 1}のような配列を用い、ai/-aiをsign[i]*aiで表すようにしてfor文を使うとよい。つまり、

$$v1=a1+\underbrace{sign[?]}_{-1,\ 1}*a2$$ のすべてと $$v2=\underbrace{sign[?]}_{-1,\ 1}*a3+\underbrace{sign[?]}_{-1,\ 1}*a4$$ のすべての組み合わせ（v1+v2）を作ればよい

- 答え合わせ：
  - 1±2±3±4≥ 2なら場合の数は4