Induction (帰納) の話
亀山

問題 1 グラフの性質

問題. 「全ての木は、サイズが位数より 1 小さい」ことを証明せよ。

解答 ?. 葉とそれにつながる辺を除去する。この時「位数-サイズ」は不変である。**それを繰返すと、いつか、**根だけからなる木になる。「位数-サイズ」は 1 である。よって、どんな木でも「位数-サイズ」は 1 である。

解答. 論理式 $P(n)$ を、「サイズ n の木は、(どんな木であっても) その頂点の数 (位数) は $n+1$ である」を表すものとする。

- $P(0)$ は成立する。
  なぜなら、サイズ 0 の木は、根だけからなる木しかなく、これの頂点の数は 1 だから。
- $n > 0$ に対して、$P(n)$ が成立すると仮定すると、$P(n+1)$ は成立する。
  なぜなら、サイズ $n+1$ の木から、(どれでもいいので) 葉を 1 つと、その葉につながっている辺 (1 本しかない) を除去すると、また木になり、また、それはサイズ n である。よって、$P(n)$ を使って、その (除去後の) 木の頂点数は $n+1$ であることがわかる。すると、除去前の木の頂点数は $n+2$ であることがわかる。よって、$P(n+1)$ が成立する。

よって、どんな自然数 n に対しても $P(n)$ が成立する ($\forall n \in \mathcal{N}.\ P(n)$)。

問題 2 アルゴリズムの計算量

ソートとよばれるアルゴリズムは、サイズ n の配列 (要素は整数とする) を入力すると、それを小さい順に並べた配列を返す。

ソート・アルゴリズムの中でも、マージソートというものでは、「サイズ n の配列のソートに要する時間」を T(n) とすると、次の不等式が成立する。

$$
\begin{aligned}
&T(1) = 0 \\
&T(n) < 2 \cdot T(\frac{n}{2}) + n \qquad\qquad \text{if } n > 1
\end{aligned}
$$

ただし、ここでは簡単のために、n は 2^k の形をしているとする。このとき、以下が成立する (ことを帰納法で証明できる)。

$$T(n) \le n \cdot k$$

$k = \log_2 n$ なので、この式は、 $T(n) \le n \log_2 n$ ということを意味していて、つまり、「マージソート・アルゴリズムの計算量は、$O(n \log n)$ だ」と言われることの根拠になっている。($O(\ )$ という記法や「計算量」という言葉については、「データ構造とアルゴリズム」の授業で勉強してほしい。)

解答. 「任意の自然数 k に対して、$T(2^k) \le k \cdot 2^k$ が成立すること」を k に関する数学的帰納法で証明すればよい。(ちなみに、「任意の自然数 k に対して、$T(2^k) < k \cdot 2^k$ が成立すること」は証明できない。)

問題 3 プログラム言語の構文

あるプログラム言語の式 (expression) は、以下の構文をもつ。

$$e ::= 0 \mid 1 \mid ... \mid 9 \mid x \mid y \mid (e+e) \mid (e*e)$$

このとき、式の (文字列としての) 長さは奇数であることを示せ。

解答. まず、「式の長さ」を帰納的に定義する。次に、「任意の式 e に対して、e の長さが奇数であること」を**式の構成に関する帰納法**で証明すればよい。

問題 4 より高度な帰納法

次のように再帰的に定義された部分関数 (partial recursive function) `gcd` が、どんな自然数に対しても値が定義されていること、つまり、`gcd`: $\mathcal{N} \to \mathcal{N}$ という関数であることを示せ。

```
gcd(x,y) =
 if x=y then x
 else if x>y then gcd(x-y,y)
 else gcd(y-x,x)
```

解答. 再帰呼び出しをするごとに、$x+y$ の値小さくなっていくことに着目して、「任意の自然数 k に対して、$x+y<k$ であば、`gcd`(x,y) の計算は有限時間が停止する (値が求まる)」ことを、k に関する数学的帰納法で証明すればよい。

次の関数 mc はどうか？

```
mc(x) =
 if x>=101 x-10
 else f(f(x+11))
```

解答. これはかなり難しい。よく考えると、x の値がだんだん大きくなって 101 に近くなると収束することがわかるので、「$101-x$ の値に関する数学的帰納法」を使えそうだ、とわかるが、これをヒントなしにわかるようになるのは、かなり修業を積む必要がある。(離散構造の授業の範囲では、そこまではやらない。)