4.再帰という考え方

再帰的定義

- 再帰(recursion)：関数や関係の定義中に、自分自身の記述が現れるもの
- 例1:フィボナッチ数列

$$\begin{cases} f(0)=1 \\ f(1)=1 \\ f(n+1)=f(n)+f(n-1) \end{cases}$$

←fの定義中にfが現れている

- 例2:階乗

$$\begin{cases} f(1)=1 \\ f(n)=n*f(n-1) \end{cases}$$

なぜ再帰を使うか

- 再帰的な構造をもつ関数の記述が簡単になる

- プログラムの見通しが良くなる
  - ある問題を解くときに、それより小さい問題の答を使って解く

プログラムにおける再帰

- 関数gの定義の中に自分自身が現れる

```
#include <stdio.h>
int g(int x){
  if (x > 100)
    return x-10;
  else
    return g(g(x+11));
}
int main(void){
  int n;
  scanf(“%d”, &n);
  printf(“%d¥n”, g(n));
}
```

分割統治(divide and conquer)

- 大きな問題を、(同じ形で簡単に解ける)部分問題に分割し、その解を組み合わせて全体の解とする

- 具体例
  - 組合せの数
  - ソート(整列)

組合せの数

- ${}_nC_k$: 異なるn個からk個を選ぶ場合の数

- 単純な場合
  - ${}_nC_n$:異なるn個からn個を選ぶ場合の数=1
  - ${}_nC_1$:異なるn個から1個を選ぶ場合の数=n
- それ以外の場合
  - ${}_nC_k$:異なるn個からk個を選ぶ→(ある1個を選んで残りn-1個からk-1個を選ぶ)または(ある1個を選ばず残りn-1個からk個を選ぶ)

$${}_nC_k = {}_{n-1}C_{k-1} + {}_{n-1}C_k$$

組合せの数のプログラム

```
#include <stdio.h>
int c(int n, int k){
  if (n == k) return 1;
  else {
    if (k == 1) return n;
    else
      return c(n-1,k-1)+c(n-1,k);
  }
}
int main(void)
{
  int x,y;
  scanf(“%d”, &x); scanf(“%d”, &y);
  printf(“%d¥n”, c(x,y));
}
```

$${}_nC_n = 1$$

$${}_nC_1 = n$$

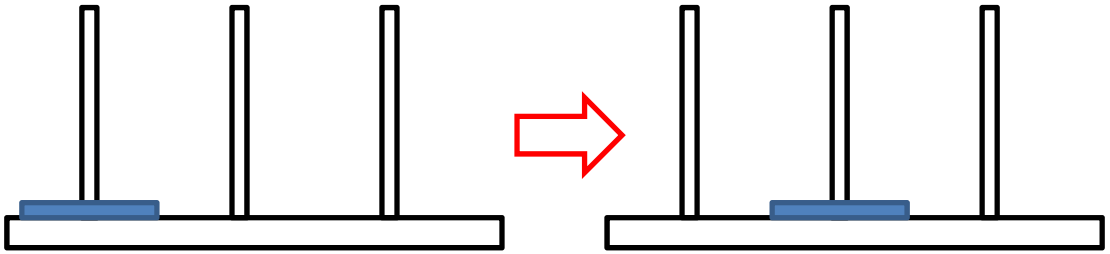
$${}_nC_k = {}_{n-1}C_{k-1} + {}_{n-1}C_k$$

ハノイの塔

- 円盤を移動するパズル
- 目標:全ての円盤を別の柱に移動
- 規則
  - (1)1度に1枚しか動かせない
  - (2)小さい円盤の上に大きい円盤を載せてはいけない
- 考えてみて下さい

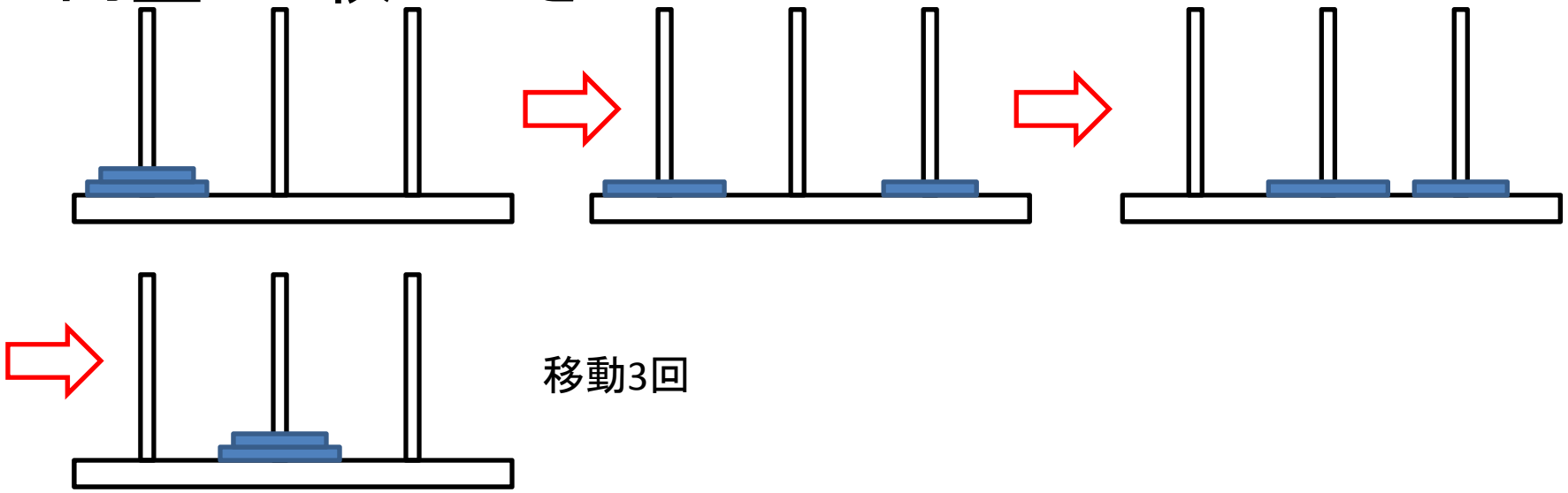
ハノイの塔:例1

- 円盤が1枚のとき:簡単



移動1回

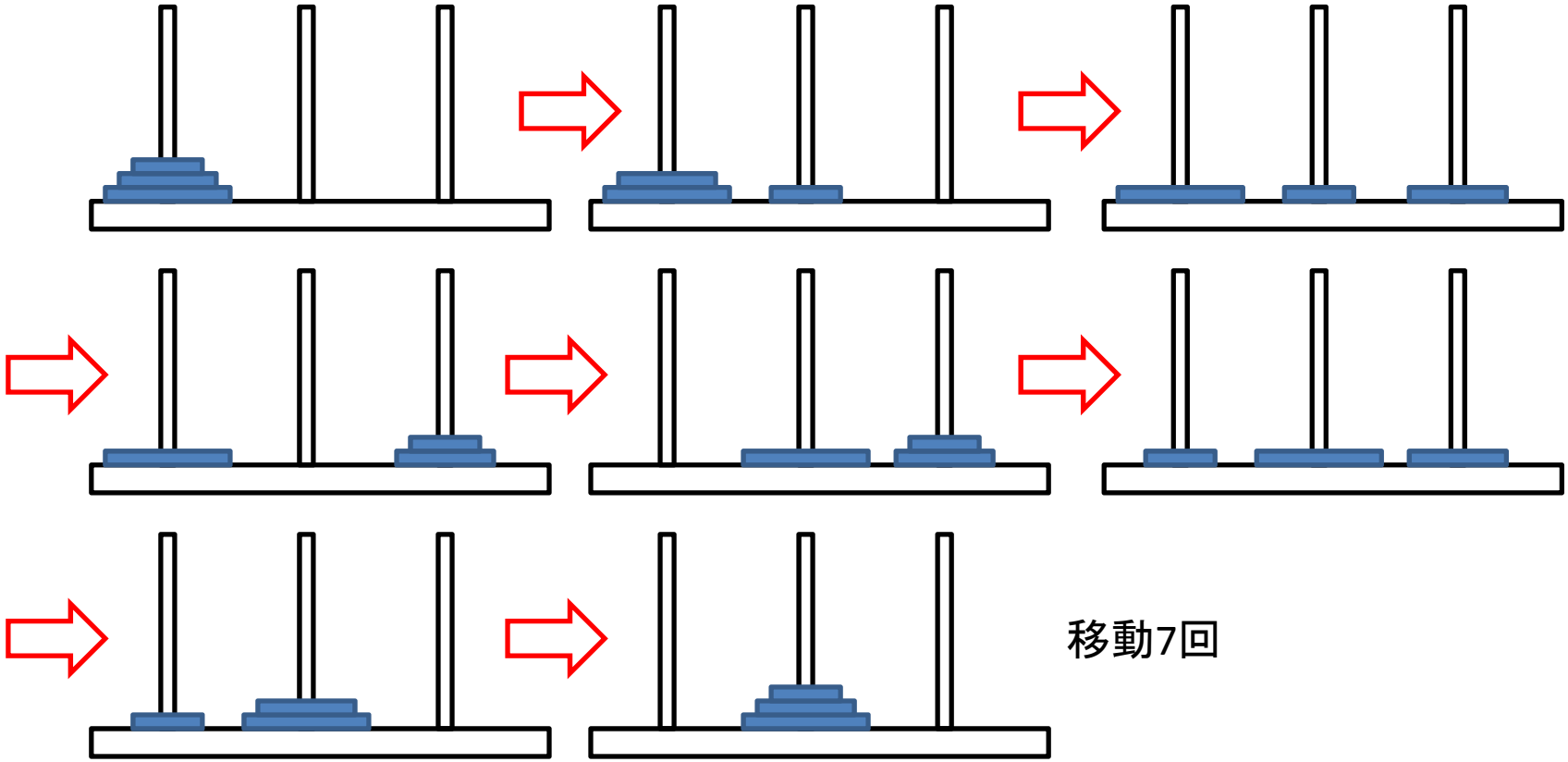
- 円盤が2枚のとき



移動3回

ハノイの塔:例2

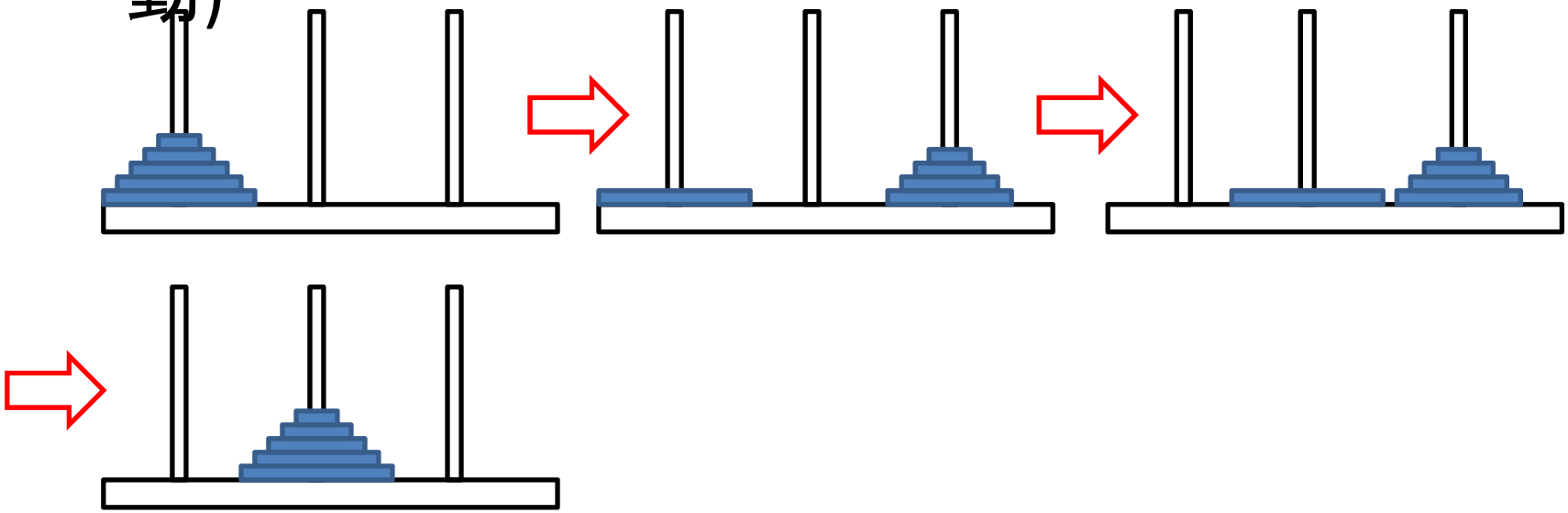
- 円盤が3枚のとき



- 円盤がn枚のときは？

円盤がn枚の場合

- n枚の移動=(最大の1枚の移動)+(n-1枚の移動)



円盤の移動の記述

- move(n,x,y,z): n枚の円盤を柱xから(柱zを使って)柱yに移動する作業
- move(1,x,y,z)は単純
- n>2の場合、move(n,x,y,z)は
  - (1)move(n-1,x,z,y) n-1枚を柱xから柱zに移動
  - (2)柱xから柱yに、最大の円盤1枚を移動
  - (3)move(n-1,z,y,x) n-1枚を柱zから柱yに移動

ハノイの塔のプログラム

```
#include <stdio.h>
void move(int n, int a, int b, int c){
  if (n == 1){
    printf("From %d to %d ¥n", a, b);
  }
  else {
    move(n-1,a,c,b);
    printf("From %d to %d ¥n", a, b);
    move(n-1,c,b,a);
  }
}
int main (void){
  int m;
  scanf("%d",&m);
  move(m,1,2,3);
  return 0;
}
```

円盤が1枚なら移動

円盤がn枚なら
①n-1枚をa→cへ移動
②1枚をa→bに移動
③n-1枚をc→bに移動

移動回数

- 円盤1枚:1回
- 円盤2枚:3回
- 円盤3枚:7回
- 円盤n枚:$\underbrace{2^0+2^1+2^2+\ldots+2^{n-1}}_{n個}$

  $=2^n-1$個??

moveの定義より
n枚の移動回数=(n-1枚の移動回数)×2+1回
→$a_n=2a_{n-1}+1$　$a_n=2^n-1$