

前回課題の解説

(第12回: 実践的応用)

担当: 佐藤

2018年12月17日(月)

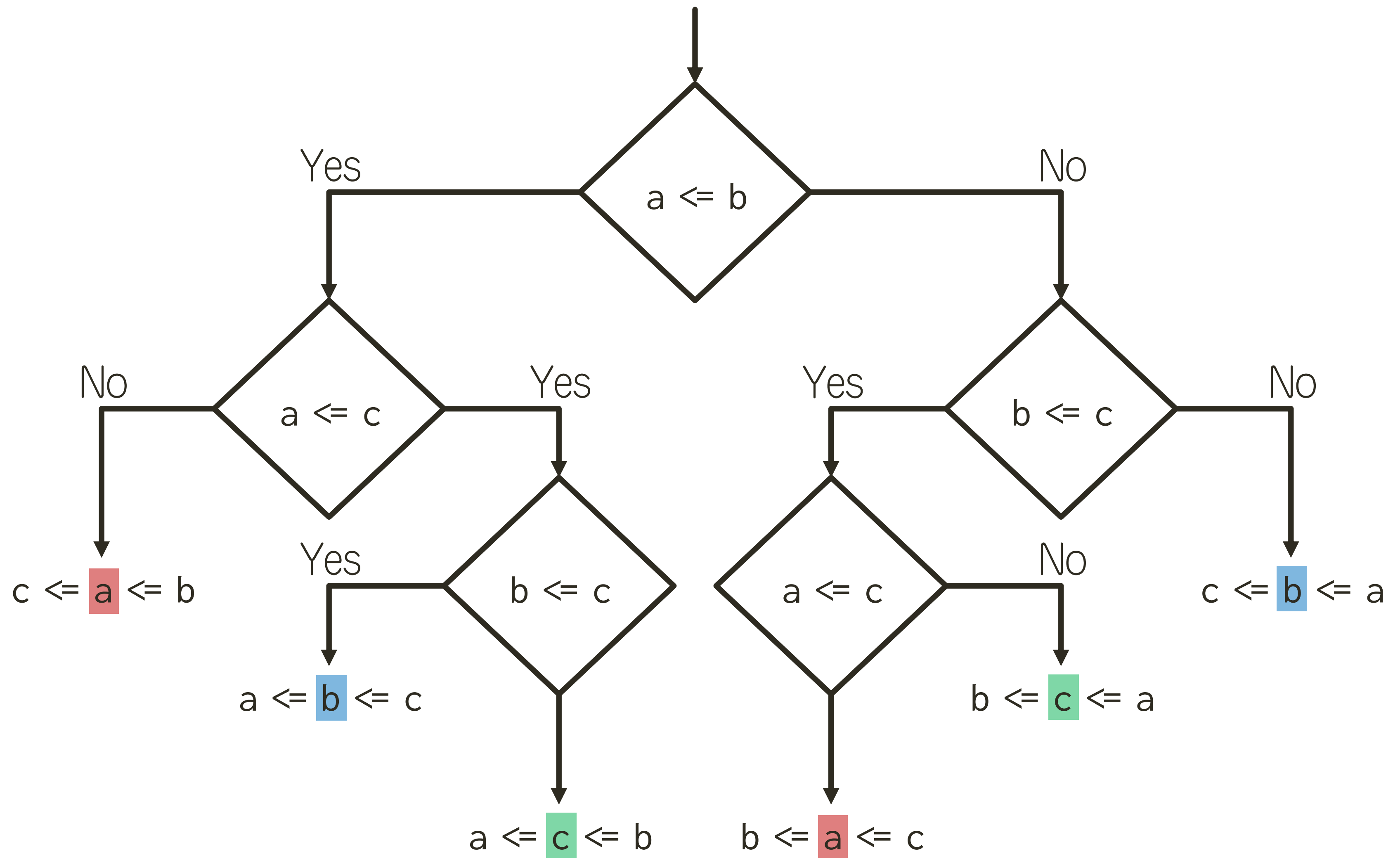
課題12-1(1)

- 3つの整数(a, b, c)を大小関係で場合分けすると $3! = 6$ 通り(※)
 - $a \leq b \leq c$
 - $a \leq c \leq b$
 - $b \leq a \leq c$
 - $b \leq c \leq a$
 - $c \leq a \leq b$
 - $c \leq b \leq a$

※ 本課題は6通りしかないので場合分けによるプログラムを書くことができる。しかし、 $n!$ は n に応じて急激に増加するため、 n が大きい場合、場合分けによるプログラムを書くことは現実的に不可能である。本課題の一般化である「 n 個のデータの中央値(メディアン)を求めるプログラム」を作成する際には、 n 個のデータを昇順/降順に並べ換える操作(整列)が必要になることを覚えておこう!

課題12-1(2)

場合分けのフローチャート(※)



※ 等号(=)の有無によって真ん中の値は変わらない→あえて等号をつけている箇所がある

課題12-1(3)

別解(5行3数値のデータ入力を前提)

```
for (i = 0; i < 5; i++) { /* iを宣言しておくこと */
    fscanf(fpr, "%d,%d,%d", &v[0], &v[1], &v[2]);
    ...
}
```

```
#include <stdio.h>

int main() {
    FILE *fpr, *fpw;
    int v[3], median;
    char fr[] = "multiRowData.csv",
        fw[] = "result.txt";

    fpr = fopen(fr, "r");
    if (fpr == NULL) {
        printf("%sをオープンできません\n", fr);
        return 1;
    }

    fpw = fopen(fw, "w");
    if (fpw == NULL) {
        printf("%sをオープンできません\n", fw);
        return 1;
    }
}
```

(右へ続く)

```
while (fscanf(fpr, "%d,%d,%d",
               &v[0], &v[1], &v[2]) != EOF) {
    if (v[0] <= v[1])
        if (v[0] <= v[2])
            if (v[1] <= v[2]) median = v[1];
            else median = v[2];
        else median = v[0];
    else
        if (v[1] <= v[2])
            if (v[0] <= v[2]) median = v[0];
            else median = v[2];
        else median = v[1];
    fprintf(fpw, "%d\n", median);
}

fclose(fpr);
fclose(fpw);

return 0;
}
```

課題12-1(4)

別解(論理演算子を利用)

- 「"%d, %d, %d"」でもよい
 - scanf()族の書式指定文字列中の空白類文字は、入力文字列における0個以上の空白類文字に対応する

```
#include <stdio.h>

int main() {
    FILE *fpr, *fpw;
    int v[3], median;
    char fr[] = "multiRowData.csv",
        fw[] = "result.txt";

    fpr = fopen(fr, "r");
    if (fpr == NULL) {
        printf("%sをオープンできません\n", fr);
        return 1;
    }

    fpw = fopen(fw, "w");
    if (fpw == NULL) {
        printf("%sをオープンできません\n", fw);
        return 1;
    }
}
```

```
while (fscanf(fpr, "%d,%d,%d",
             &v[0], &v[1], &v[2]) != EOF) {
    if (v[1] <= v[0] && v[0] <= v[2] ||
        v[2] <= v[0] && v[0] <= v[1])
        median = v[0];
    if (v[0] <= v[1] && v[1] <= v[2] ||
        v[2] <= v[1] && v[1] <= v[0])
        median = v[1];
    if (v[0] <= v[2] && v[2] <= v[1] ||
        v[1] <= v[2] && v[2] <= v[0])
        median = v[2];
    fprintf(fpw, "%d\n", median);
}

fclose(fpr);
fclose(fpw);

return 0;
}
```

(右へ続く)

課題12-1(5)

別解(調べる場合を半分にする)

```
#include <stdio.h>
```

```
int main() {  
    FILE *fpr, *fpw;  
    int v[3], median, tmp;  
    char fr[] = "multiRowData.csv",  
         fw[] = "result.txt";  
  
    fpr = fopen(fr, "r");  
    if (fpr == NULL) {  
        printf("%sをオープンできません\n", fr);  
        return 1;  
    }  
  
    fpw = fopen(fw, "w");  
    if (fpw == NULL) {  
        printf("%sをオープンできません\n", fw);  
        return 1;  
    }  
}
```

最初に2つの値の
大小を決めておく

```
while (fscanf(fpr, "%d,%d,%d",  
              &v[0], &v[1], &v[2]) != EOF) {  
    if (v[0] < v[1]) {  
        tmp = v[0]; v[0] = v[1]; v[1] = tmp;  
    }  
    if (v[1] > v[2]) median = v[1];  
    else if (v[0] > v[2]) median = v[2];  
    else median = v[0];  
    fprintf(fpw, "%d\n", median);  
}  
  
fclose(fpr);  
fclose(fpw);  
  
return 0;  
}
```

(右へ続く)

課題12-2(1)

$f(x)$ と $f'(x)$ をあらかじめ
自作関数として定義

α は $f()$ と $\text{main}()$
の両方で用いる
ため、グローバル
変数として宣
言しておく

```
#include <stdio.h>
#include <math.h>

double alpha;

double f(double x) {
    return x * x - alpha;
}

double df(double x) {
    return 2 * x - 1;
}
```

(右へ続く)

問題文の指示にしたが
い ε と x に初期値を設定

```
int main() {
    double EPS = pow(10, -6);
    double x1, x2 = 1.0;

    scanf("%lf", &alpha);
    while (1) {
        x1 = x2;
        x2 = x1 - f(x1) / df(x1);
        if (fabs(x2 - x1) < EPS) break;
    }
    printf("%fの平方根は%f¥n", alpha, x2);

    return 0;
}
```

収束条件を満たすまで
漸化式で x の値を更新

課題12-2(2)

別解(数学関数を使わず関数定義もしない場合)

```
#include <stdio.h>

int main() {
    const double EPS = 1e-6;
    double x1, x2 = 1.0, alpha;

    scanf("%lf", &alpha);
    while (1) {
        x1 = x2;
        x2 = (x1 + alpha / x1) / 2;
        if (x1 < x2) {
            if (x2 - x1 < EPS) break;
        } else {
            if (x1 - x2 < EPS) break;
        }
    }
    printf("%gの平方根は%f¥n", alpha, x2);

    return 0;
}
```

$$x_{n+1} = -\frac{f(x_n)}{f'(x_n)} + x_n$$

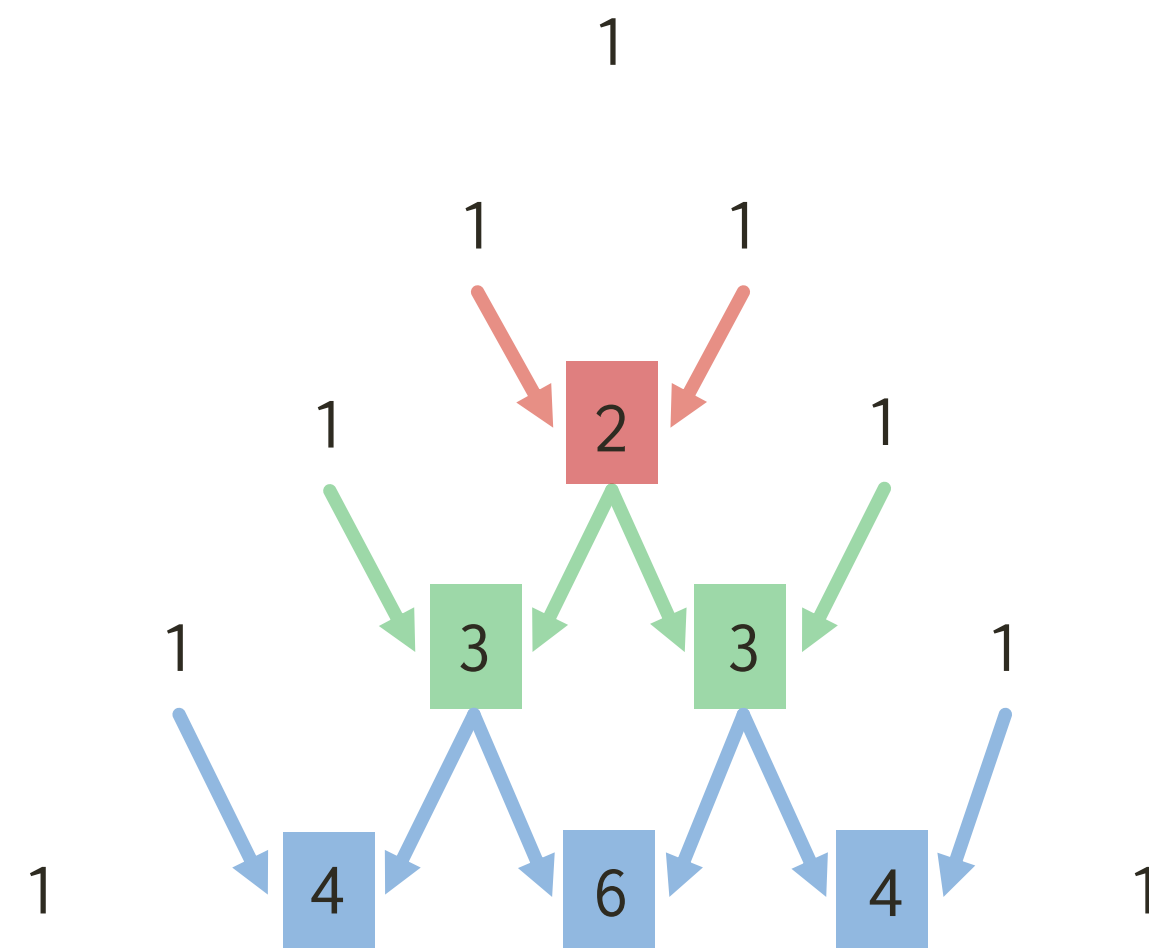
に $f(x_n) = x_n^2 - \alpha$, $f'(x_n) = 2x_n$ を代入

課題12-3(1)

- step1: 段ごとに配列を用意し, 規則性を反映して値を設定
 - どの段も左端と右端の値は1
 - 3段目以降 → 左端と右端以外の値はひとつ前の段の左右の値の和

```
int l1[1], l2[2], l3[3], l4[4], l5[5];
int i, j;

l1[0] = 1;
l2[0] = l2[1] = 1;
l3[0] = l3[2] = 1;
for (j = 1; j > 0; j--) {
    l3[j] = l2[j] + l2[j - 1];
}
l4[0] = l4[3] = 1;
for (j = 2; j > 0; j--) {
    l4[j] = l3[j] + l3[j - 1];
}
l5[0] = l5[4] = 1;
for (j = 3; j > 0; j--) {
    l5[j] = l4[j] + l4[j - 1];
}
```

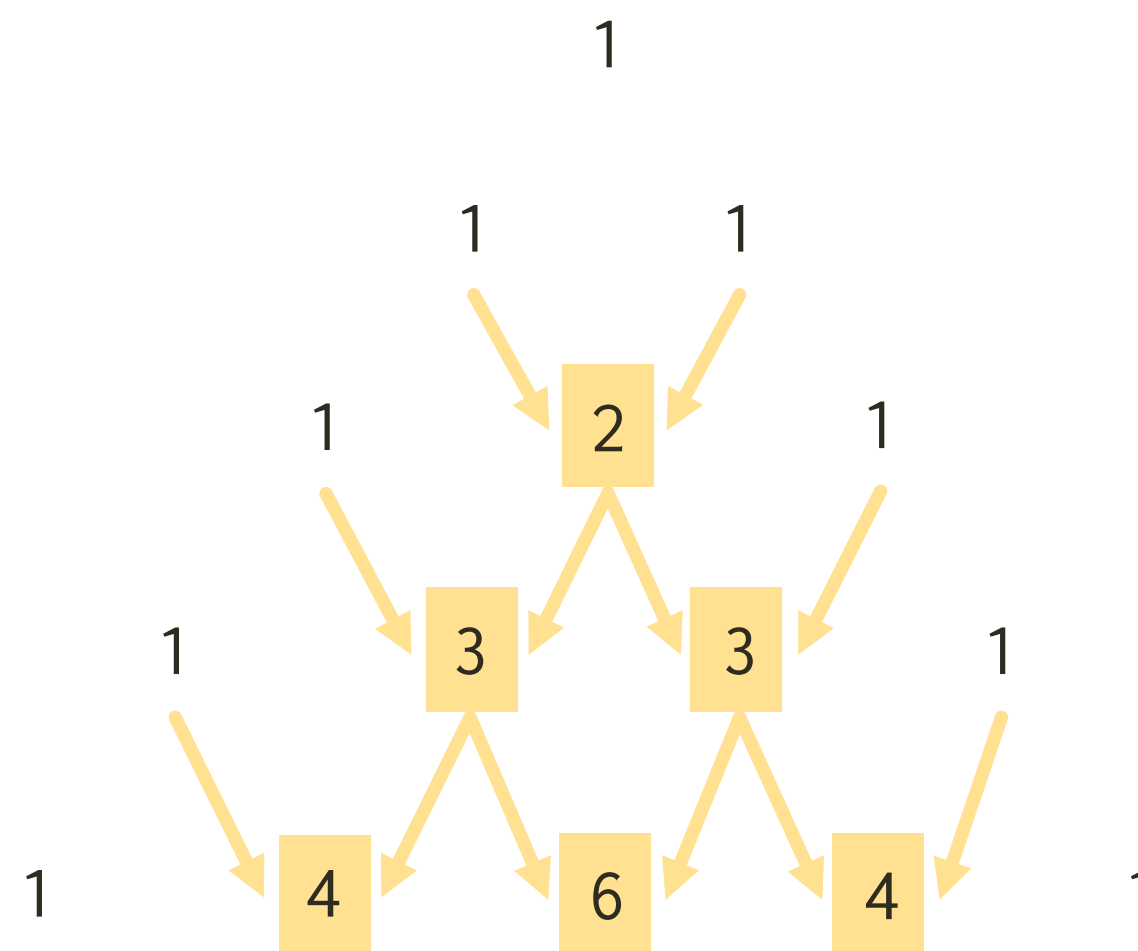


課題12-3(2)

- step2: 1つの配列で複数の段を表現できるよう変更
 - for文の中で「各段の値の設定→表示」を繰り返すことで可能

```
int a[5];
int i, j;

for (i = 0; i < 5; i++) {
    a[i] = 1;
    for (j = i - 1; j > 0; j--) {
        a[j] = a[j] + a[j - 1];
    }
    for (j = 0; j <= i; j++) {
        printf("%2d", a[j]);
    }
    printf("\n");
}
```



実行結果

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
```

課題12-3(3)

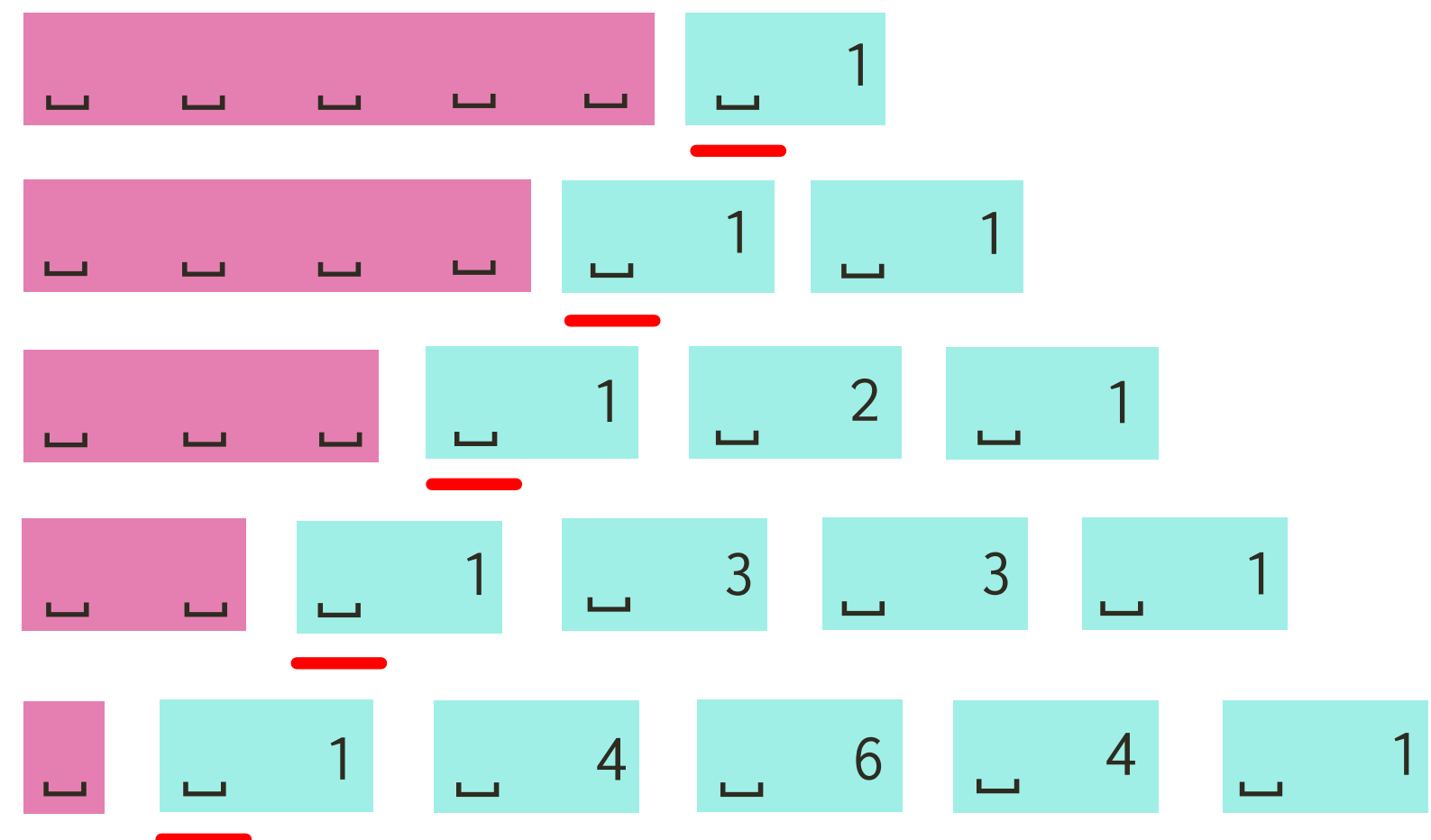
- step3: 各段の値を規則性を反映して表示

- 各段のすべての値の前には空白がある
- 画面左端から各段の左端の値が表示されるまでに挿入される空白の数には次の規則性がある

- 画面左端から各段の左端の値までの空白の数 == 段数 - 段番号 + 1

```
int a[5];
int i, j;

for (i = 0; i < 5; i++) {
    a[i] = 1;
    for (j = i - 1; j > 0; j--) {
        a[j] = a[j] + a[j - 1];
    }
    for (j = 0; j < 5 - i; j++) {
        printf(" ");
    }
    for (j = 0; j <= i; j++) {
        printf("%2d", a[j]);
    }
    printf("\n");
}
```



課題12-3(4)

- step4: 段数はmain()から受け取るようにしてdrawPT()を作成

```
#include <stdio.h>

void drawPT(int n) {
    int a[5], i, j;

    for (i = 0; i < n; i++) {
        a[i] = 1;
        for (j = i - 1; j > 0; j--) {
            a[j] += a[j - 1];
        }
        for (j = 0; j < n - i; j++) {
            printf(" ");
        }
        for (j = 0; j <= i; j++) {
            printf("%2d", a[j]);
        }
        printf("¥n");
    }
}
```

(右へ続く)

```
int main() {
    int n;

    printf("1以上5以下の整数を入力してください¥n");
    scanf("%d", &n);
    drawPT(n);

    return 0;
}
```