

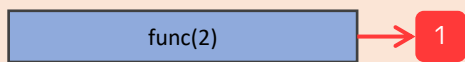
問題 3.6.1

答えは以下の通りになります。特に `func(2)` では最初の条件分岐 ($N \leq 2$) で値が返されることに注意してください。

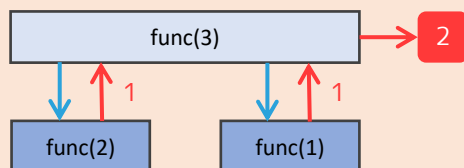
関数の呼び出し	<code>func(2)</code>	<code>func(3)</code>	<code>func(4)</code>	<code>func(5)</code>
答え	1	2	3	5

再帰呼び出しのイメージ図は以下のようになります。なお、関数 `func(N)` はフィボナッチ数 (→3.7.2項) の第 N 項を返します。

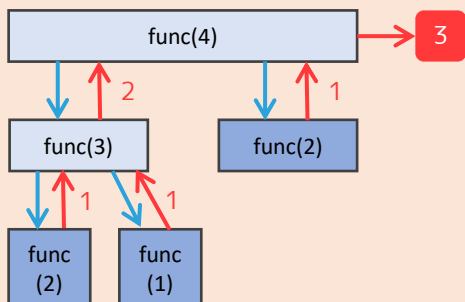
func(2) の呼び出し



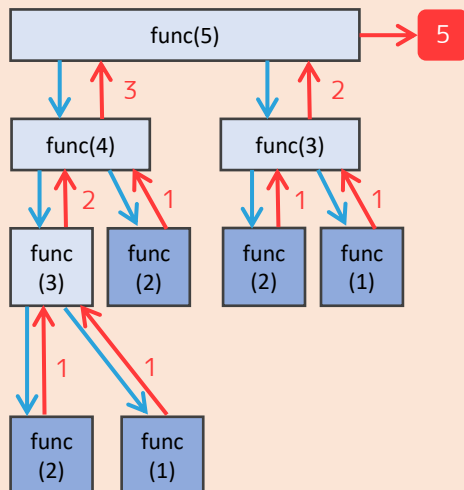
func(3) の呼び出し



func(4) の呼び出し



func(5) の呼び出し



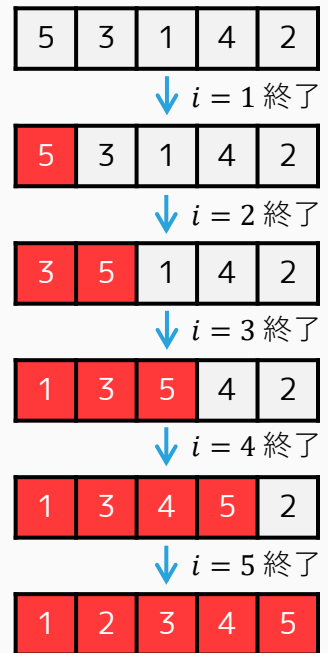
問題 3.6.2

このソートアルゴリズムでは、以下の事実 A が成り立ちます。右図は $A = [5, 3, 1, 4, 2]$ の場合の例を示しています。

$i = I$ のループが終わった時点で、
 $A[1] \leq A[2] \leq A[3] \leq \dots \leq A[I]$ が成り立つ。

これは次のように証明することができます。

なお、「 $i = I - 1$ で条件を満たすと仮定して、 $i = I$ で条件を満たすことを示す」という証明技法を **数学的帰納法** といいます。



証明したいこと (事実 B)

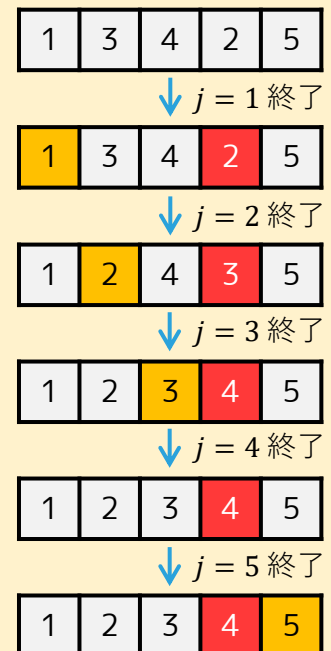
$i = I - 1$ のループが終わった時点で $A[1] \leq A[2] \leq \dots \leq A[I - 1]$ を満たす時、
 $i = I$ のループをすると $A[1] \leq A[2] \leq \dots \leq A[I]$ を満たすようになる。

事実 B の証明

$i = I - 1$ の時点で、 $A[t - 1] \leq A[I] \leq A[t]$ のとき

- $j = 1, \dots, t - 1$ では swap されない。
- $j = t$ で初めて swap される。
- $j = t + 1, \dots, I - 1$ でも swap される。
- それ以降: $A[I]$ の値が増える可能性はあるが、減ることはない

となるため、最終的には $A[1] \leq A[2] \leq \dots \leq A[I]$ が成り立ちます。右図はその一例です。



事実 B が証明できたら？

明らかに $i = 1$ のとき事実 A が成り立ちます。また、事実 B より、 $i = 2$ のとき事実 A が成り立ちます。事実 B を繰り返し適用させると、 $i = N$ のときも事実 A が成り立つ（操作終了時にソートされている）と分かります。

問題 3.6.3

まず、列 B' が空の場合、列 A' の最も左の要素 **A[c1]** を取り出すため、以下のよう
なプログラムを書くと良いです。ここで、要素を取り出した後、列 A' の最も左の位
置 **c1** が 1 増えることに注意してください。

```
else if (c2 == r) {  
    // 列 B' が空の場合  
    C[cnt] = A[c1]; c1++;  
}
```

次に、列 A'・列 B' 両方が空でない場合、以下のように場合分けすると良いです。

- 列 A' の左端 **A[c1]** が列 B' の左端 **A[c2]** より小さい：**A[c1]** を取り出す
- 列 B' の左端 **A[c2]** が列 A' の左端 **A[c1]** より小さい：**A[c2]** を取り出す

これをプログラムにすると、以下のようになります。

```
else if {  
    // 列 B' が空の場合  
    if (A[c1] <= A[c2]) {  
        C[cnt] = A[c1]; c1++;  
    }  
    else {  
        C[cnt] = A[c2]; c2++;  
    }  
}
```

全体をまとめた C++ のプログラム、Python・JAVA・C での解答例については、
chap3-6.md をご覧ください。