（発展）課題3

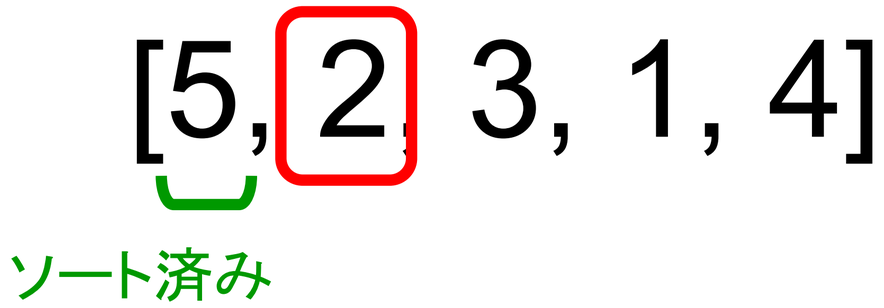
- 1-1. 挿入ソートinsertion_sort()を実装しよう。
- 1-2. 比較回数と交換回数を調べて考察してください。

挿入ソート(Insertion Sort)とは

- リストの先頭がソート済みであると見なし、残りの要素を適切な場所に挿入することで順番に並び替えるアルゴリズム
- 特徴
  - 比較回数はn(n-1) / 2回以下，計算量は$O(n^2)$

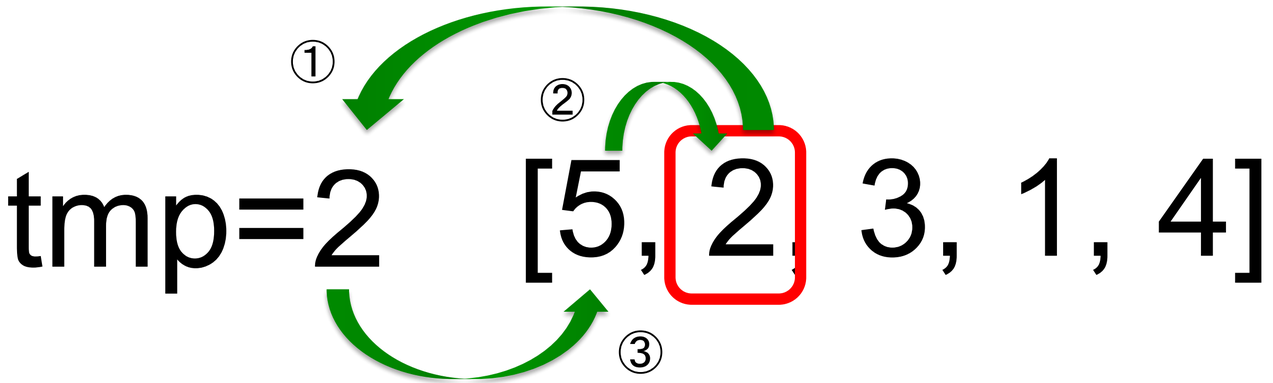
挿入ソートの解説: 先頭をソート済みと見なす

- 挿入ソートでは、ソート済みのリストに対して、追加要素をひとつずつ適切な位置に挿入します。
- 下の図は、先頭の要素（5）だけがソート済みであり、次の（2）を挿入しようとしています。
  - 先頭部分を「要素一つだけのリスト」と考えてください



挿入ソートの解説: 適切な位置に要素を挿入

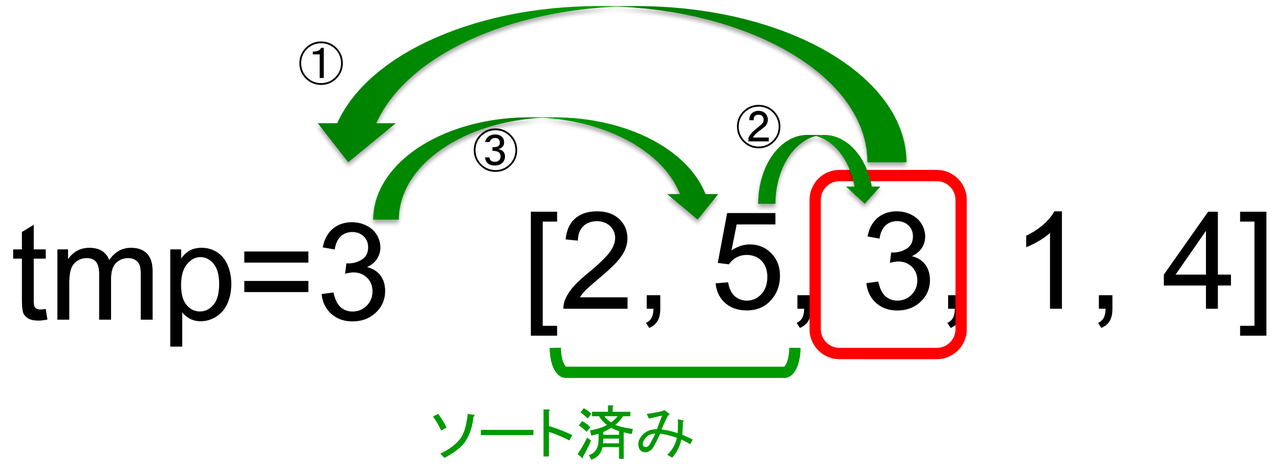
- ソート済みのリストを後ろからたどり、挿入すべき位置より後ろの要素を1つずつずらします（内側のループ）。
- 図では、(2)を一旦変数に退避し、(5)を後ろに移動させたあと、(2)を挿入しています。



“2”を”5”の前に挿入

挿入ソートの解説: ループの続き

- 先頭2つの要素がソート済みになりました。
- 残りの要素も同様に処理します。



ヒント（実装例）

```
def insertion_sort(xs):
    for i in range(1, len(xs)):
        tmp = [redacted]
        j = [redacted]
        while j >= 0 and xs[j] > [redacted]:
            [redacted]
        xs[j+1] = [redacted]

b = [5,2,3,1,4]
print("Before sort:", b)
insertion_sort(b)
print("After sort:", b)
```

```
Before sort: [5, 2, 3, 1, 4]
After sort: [1, 2, 3, 4, 5]
```