

マージソート実装の完全解剖

Pythonコードから読み解く連結リストの分割と併合

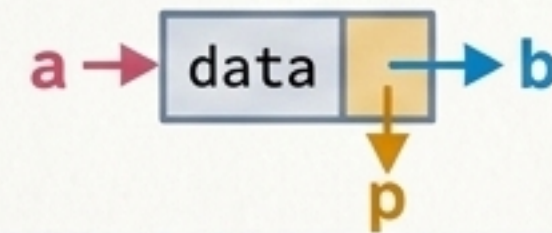
The Algorithmic Blueprint

merge_sort (オーケストレーター)

divide (分割モジュール)

merge (併合モジュール)

Cell (データ構造)



アルゴリズム全体は **merge_sort** によって統括され、未知の長さのリストを **divide** で最小単位まで分割し、**merge** で昇順に再構築する。

The Algorithmic Blueprint

```
class Cell:  
    def __init__(self, value):  
        self.value = value  
        self.next = None
```

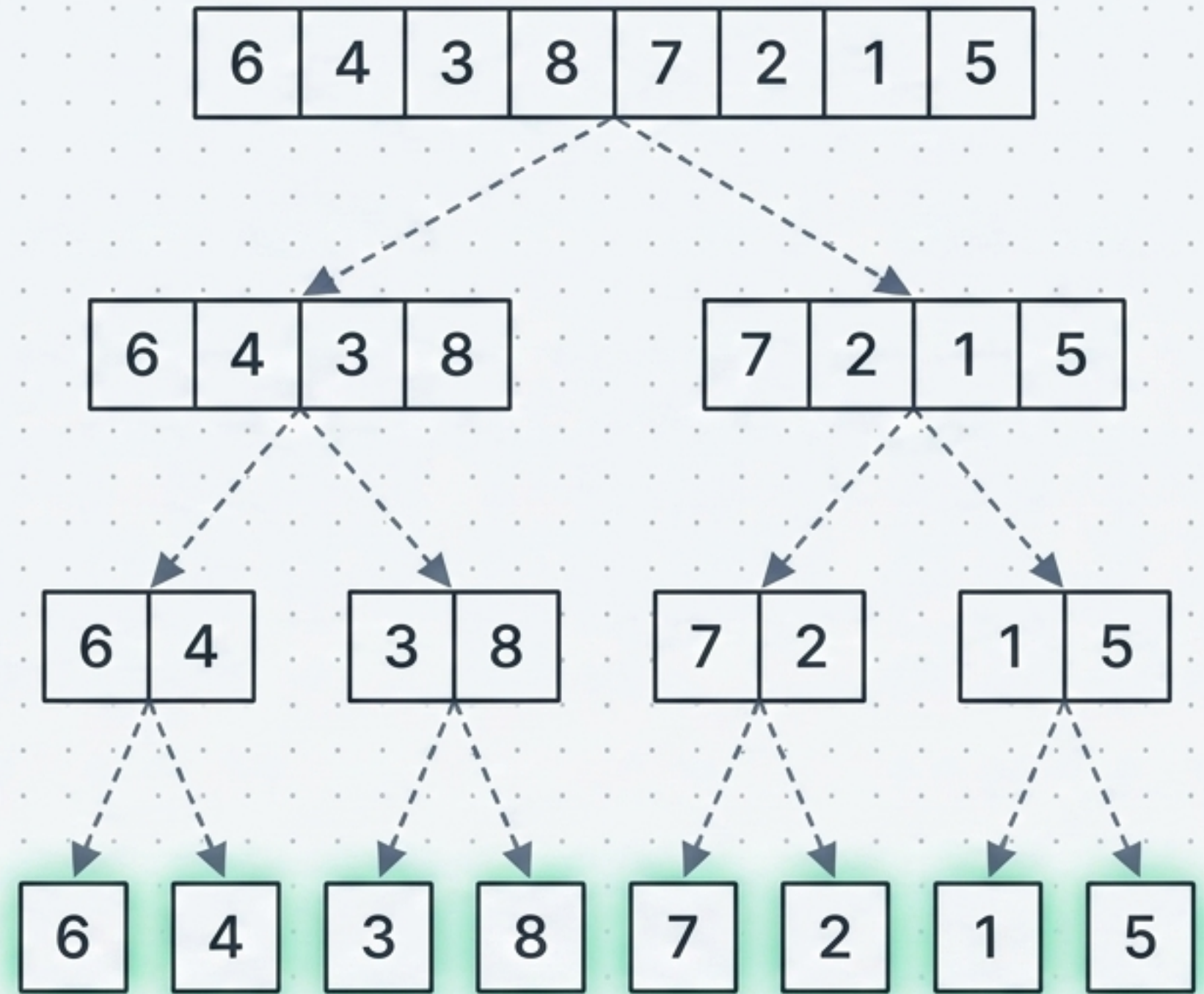


連結リストの基礎。各セルは自身の「データ (value)」と「次のセルへの参照 (next)」のみを持つ。最後尾は必ず None (NULL) となる。

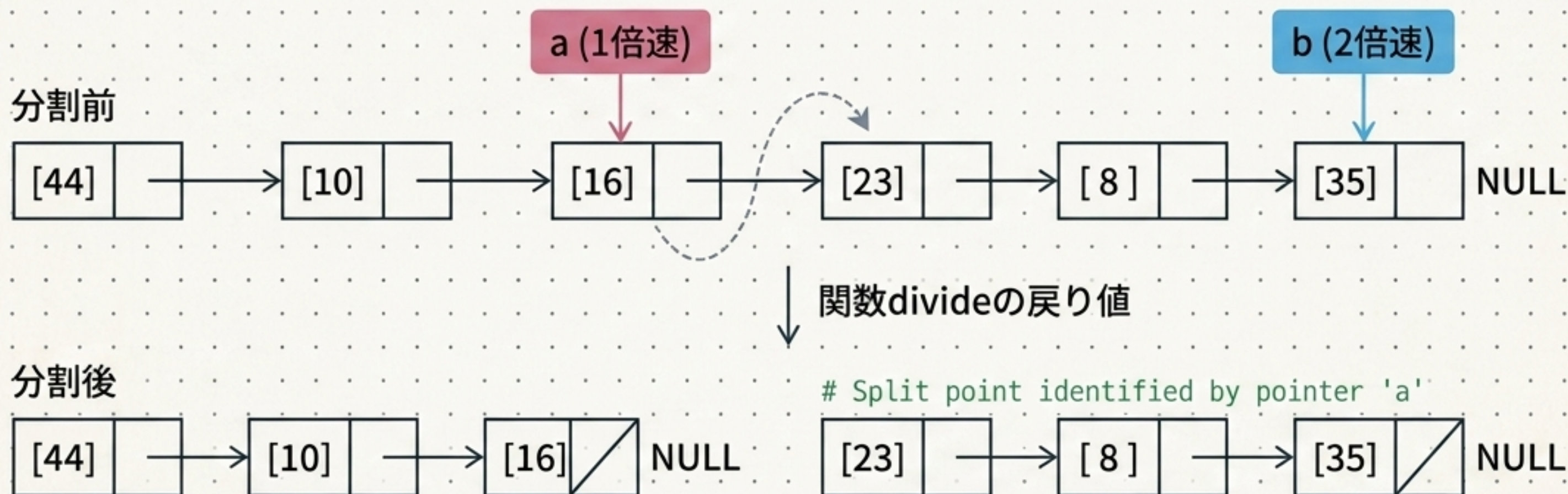
The Algorithmic Blueprint

```
def merge_sort(list_head):  
    if list_head is None or list_head.next is None:  
        return list_head  
    # ... recursive calls follow ...
```

再帰の停止条件。与えられたデータ列の要素数が1以下であれば、それ以上分割できない「整列済みのデータ列」とみなし、そのまま呼び出し元に返す。



「ウサギとカメ」のアルゴリズム。要素数が不明な連結リストを半分に分割するため、2つのポインタを使用する。ポインタ b がリストの終端に達したとき、半分の速度で進むポインタ a は必然的にリストの中央を指す。



The Algorithmic Blueprint

```
while b is not None:
```

```
    a = a.next
```

```
    b = b.next
```

```
    if b is not None:
```

```
        b = b.next
```

```
p = a.next
```

```
a.next = None
```

[ア] 停止条件

b が末尾(None)に達した瞬間、ループを終了する。

[イ] 先読み確認

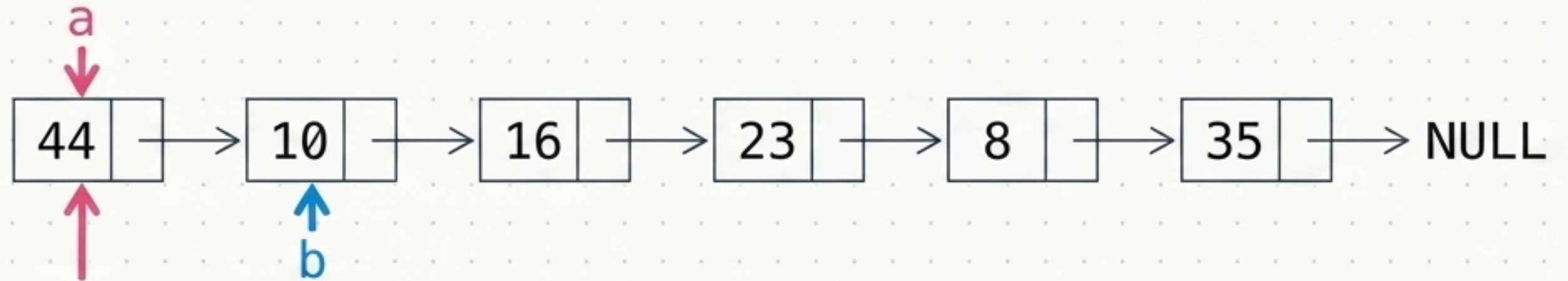
b は2歩進むため、1歩進んだ後に次の要素が存在するか必ず確認する。

[ウ] リストの切断

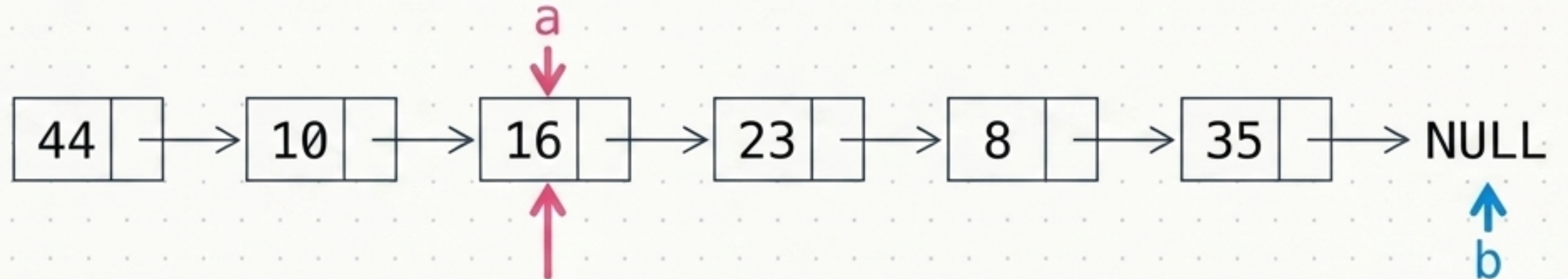
前半リストの最後にNone を代入し、連結を物理的に断ち切る。ここで分割が完成する。

The Algorithmic Blueprint

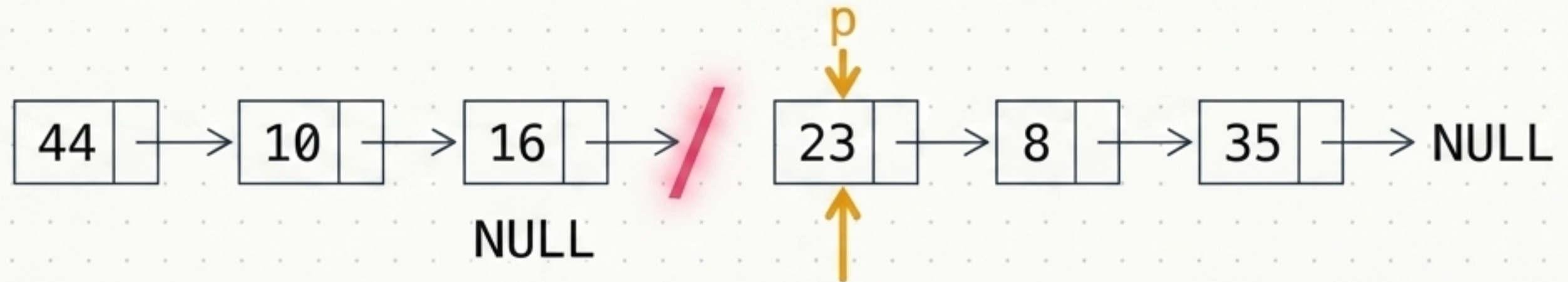
Phase 1: 初期



Phase 2: ループ完了



Phase 3: 分割後



ループ終了時、a は前半の末尾に到達。その後 `a.next = None` でリストが真っ二つに分かれる。

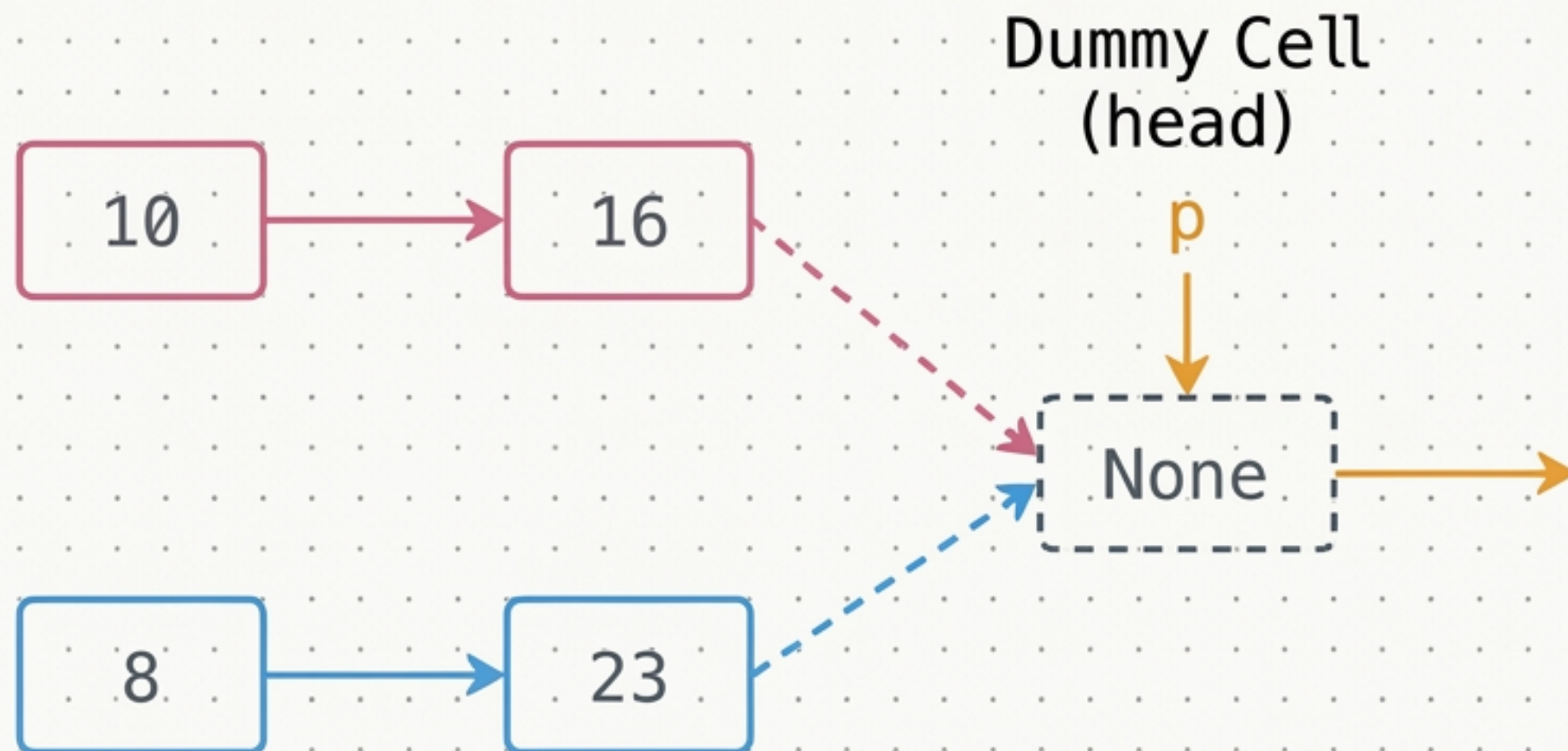
The Algorithmic Blueprint

併合（マージ）のプロセス。

2つの整列済みリストを昇順に一つにまとめる。

ゼロからリストを構築する際、起点となる「値を持たないダミーのセル(head)」を用意するこ

先頭要素の処理を一般化し、コードを簡潔に保つ。

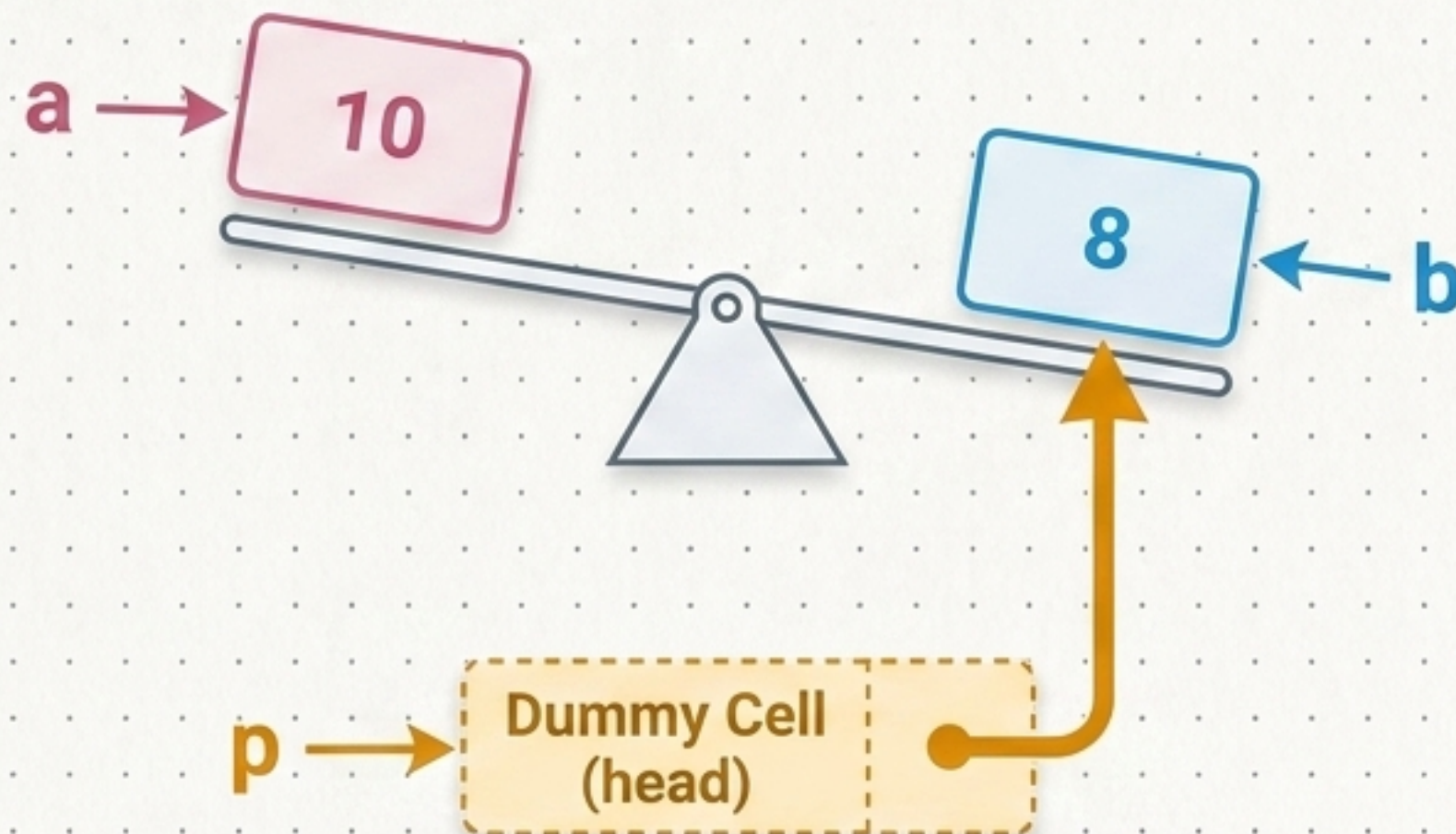


The Algorithmic Blueprint

[エ] 比較継続の条件

どちらか一方が空(None)になると比較不能になるため、「両方に要素が残っている間」だけループを回す。

```
while a is not None and b is not None:  
    if a.value <= b.value:  
        p.next = a  
        # ... pointers advance ...
```



The Algorithmic Blueprint

```
if b is not None:  
    p.next = b  
else:  
    p.next = a  
return head.next
```

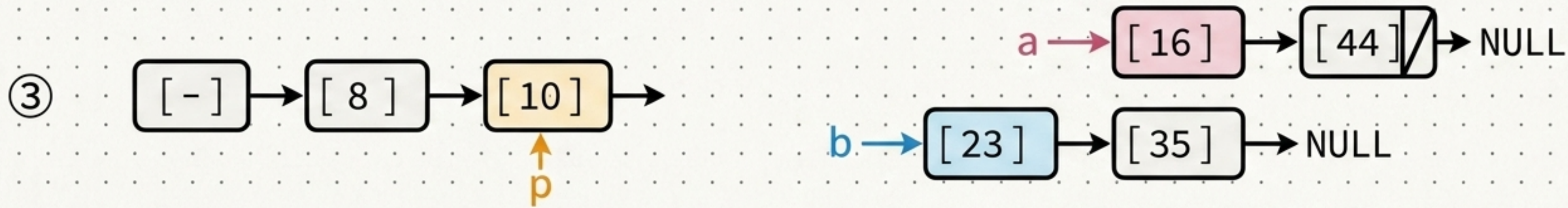
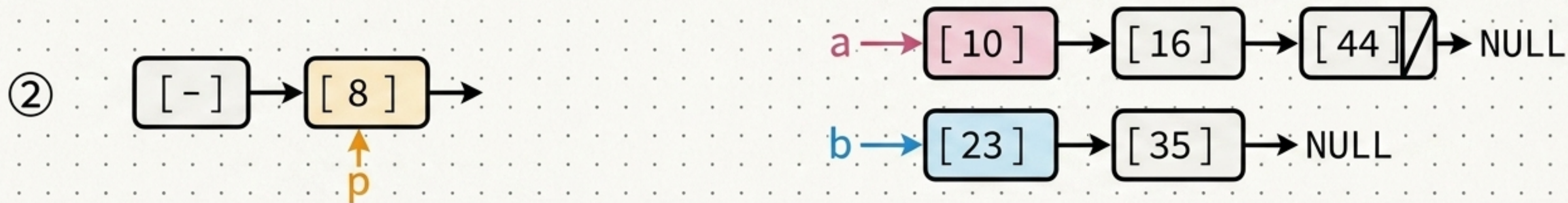
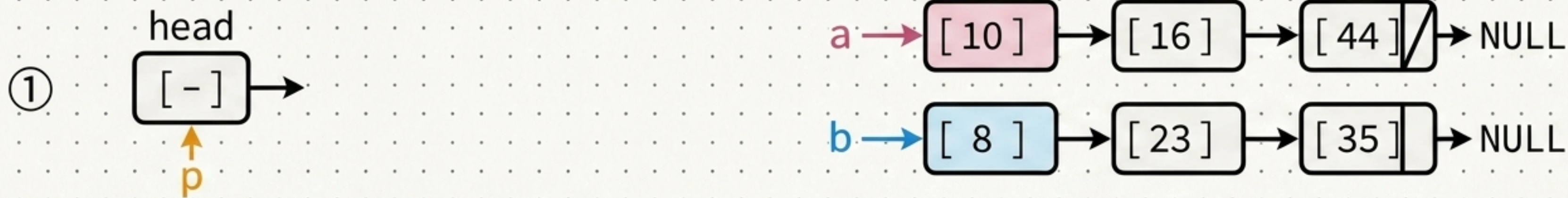
【オ】 残存データの連結

ループを抜けた時点で、空になっていない方のリストをそのまま末尾に繋ぐ。(bに要素が残っていればbを、そうでなければaを繋ぐ)。

【カ】 戻り値

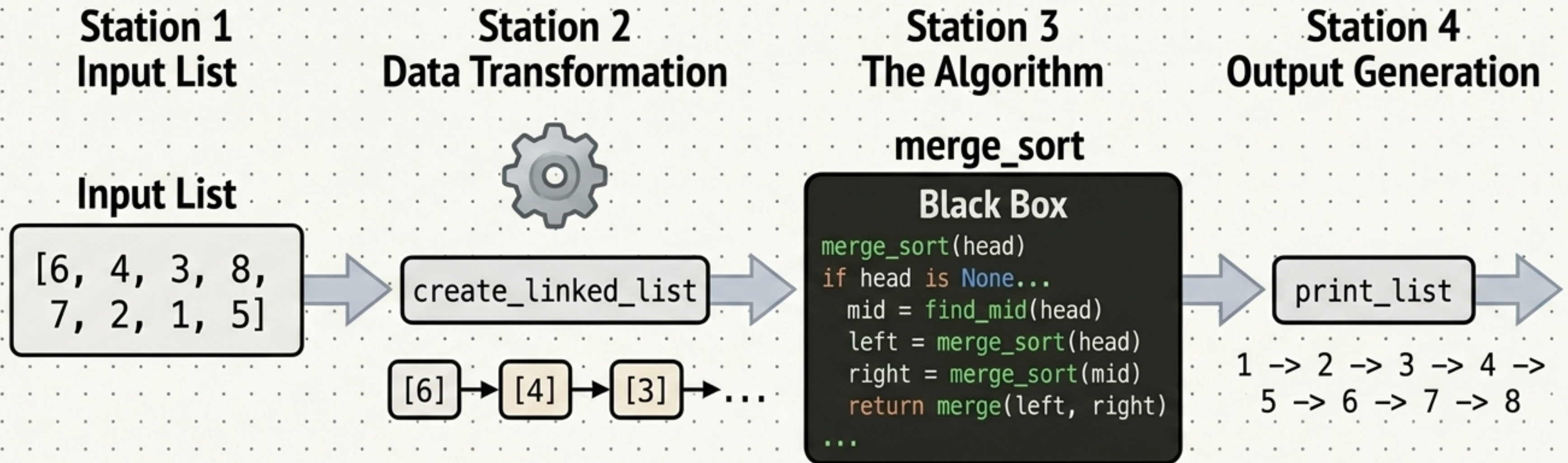
先頭の head は値を持たないダミーセルのため、その「次 (next)」にある実際のデータ群の先頭を呼び出し元へ返す。

The Algorithmic Blueprint



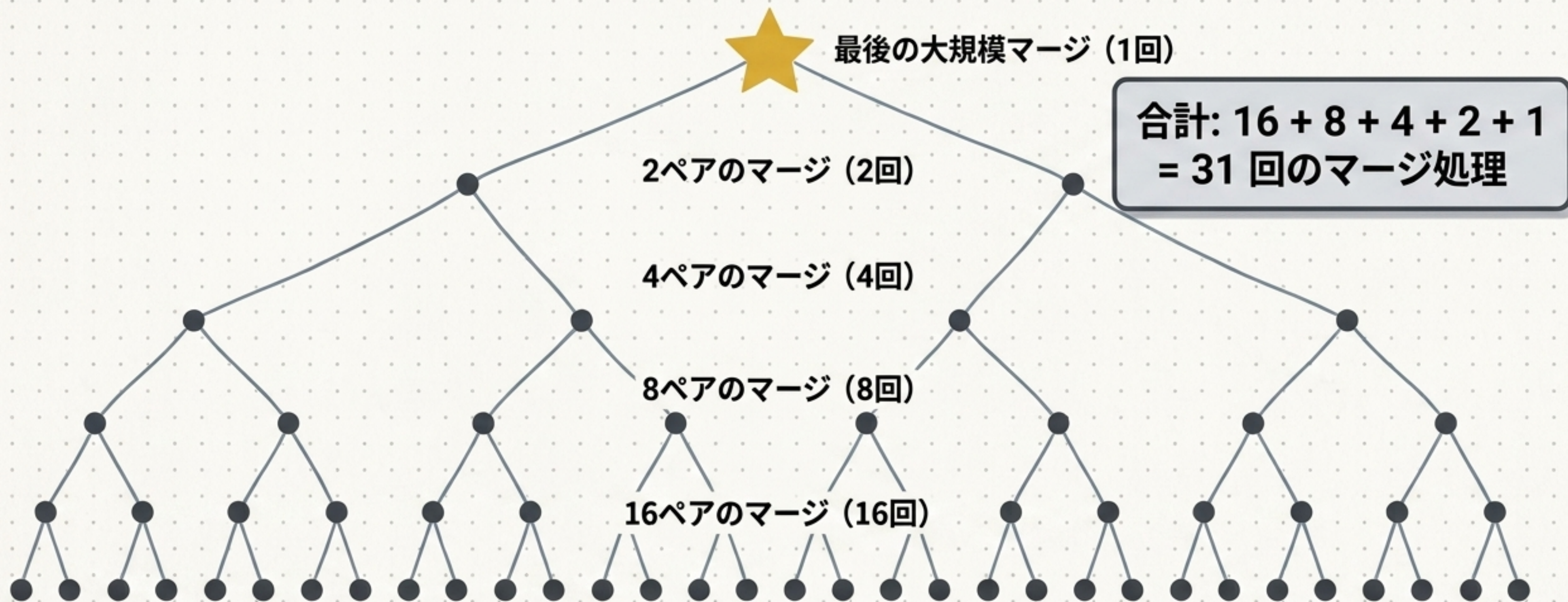
ポインタ p が進みながら、a と b の小さい方を次々と選び取り、新しい鎖を編み上げていく。

The Algorithmic Blueprint



実際の運用では、配列(List)から連結リストを生成するヘルパー関数 create_linked_list を経由してアルゴリズムにデータを流し込む。

The Algorithmic Blueprint



分割統治法の美しさ。32個のバラバラな要素も、トーナメント戦のように下から合体(マーヅ)させていくことで、わずか31回の処理で完全に整列する。Pythonコードに記されたわずかな行数が、巨大なデータに対する効率的な秩序構築を実現している。