

```

def heapify(arr, n, i):
    largest = i # 親ノードを最大値として初期化
    left = 2 * i + 1 # 左の子ノードのインデックス
    right = 2 * i + 2 # 右の子ノードのインデックス

    # 左の子ノードが親ノードより大きい場合、最大値を更新
    if left < n and arr[left] > arr[largest]:
        largest = left

    # 右の子ノードが親ノードより大きい場合、最大値を更新
    if right < n and arr[right] > arr[largest]:
        largest = right

    # 最大値が親ノードでない場合、親ノードと最大値を入れ替えて再帰的にheapifyを
    # 呼び出す
    if largest != i:
        arr[i], arr[largest] = arr[largest], arr[i]
        heapify(arr, n, largest)

def heapsort(arr):
    n = len(arr)

    # ヒープ木を構築 (最初に一度、配列をヒープに変換)
    for i in range(n // 2 - 1, -1, -1):
        heapify(arr, n, i)

    # 配列をヒープから取り出し、ソートを行う
    for i in range(n - 1, 0, -1):
        arr[i], arr[0] = arr[0], arr[i] # 最大値を配列の末尾に移動
        heapify(arr, i, 0) # ヒープのサイズを減らして再度ヒープ化

# ソートするデータを用意
data = [64, 34, 25, 12, 22, 11, 90]
print("Before sorting: ", data)
heapsort(data)
print("After sorting: ", data)

```