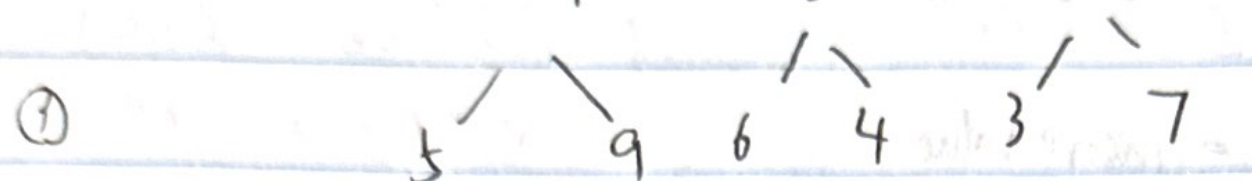


MergeSort 归并排序

5 9 6 4 3 7 1

① $\frac{0+6}{2} = 3$ 分成 5 9 6 4 3 7 1
 $\frac{4-6}{2} = 5$

② $\frac{0+3}{2} = 1$ 分成 5 9 6 4 3 7 1
 $\frac{4+6}{2} = 5$



④ 5 < 9 6 > 4 3 < 7
 合成 5 9 4 6 3 7 1

合成 4 5 6 9 1 3 7

合成 1 3 4 5 6 7 9

average $O(n \log n)$

best $O(n \log n)$

worst $O(n \log n)$

```
public static void mergesort(int[] arr, int left, int right, int[] temp)
```

```
{
    if (left < right)
    {
        int mid = (left + right) / 2;
        mergesort(arr, left, mid, temp);
        mergesort(arr, mid + 1, right, temp);
        merge(arr, left, mid, right, temp);
    }
}
```

```
public static void merge(int[] arr, int left, int right, int[] temp)
```

```
{
    int i = left;
    int j = mid + 1;
    int t = 0; // 临时数组的下标
```

```
while (i <= mid && j <= right)
{
    if (arr[i] <= arr[j])
        temp[t] = arr[i];
        i++; t++;
    else
        temp[t] = arr[j];
        j++; t++;
}
```

```
while (i <= mid) // 左边没遍历完
{
    temp[t] = arr[i];
    i++; t++;
}
```

```
while (j <= right)
{
    temp[t] = arr[j];
    j++; t++;
}
int templeft = left;
int t = 0;
while (templeft <= right)
{
    arr[templeft] = temp[t];
    templeft++; t++;
}
```