

排序

- 比较型排序
- 非比较型排序

冒泡

- 每次冒一个最大的元素到最后面去

```
void bubble_sort(int arr[], int len) {  
    int i, j, temp;  
    for (i = 0; i < len - 1; i++)  
        for (j = 0; j < len - 1 - i; j++)  
            if (arr[j] > arr[j + 1]) {  
                temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
}
```

选择

- 每次从未排序序列中选择一个最小的元素放到已排序序列的末尾

```
void swap(int *a,int *b) //交換兩個變數
{
    int temp = *a;
    *a = *b;
    *b = temp;
}
void selection_sort(int arr[], int len)
{
    int i,j;

    for (i = 0 ; i < len - 1 ; i++)
    {
        int min = i;
        for (j = i + 1; j < len; j++)    //走訪未排序的元素
            if (arr[j] < arr[min])    //找到目前最小值
                min = j;    //紀錄最小值
        swap(&arr[min], &arr[i]);    //做交換
    }
}
```

插入

- 每次将一个元素插入到合适的位置，右边的数都往后挪一个位置

```
1 void insertion_sort(int arr[], int len){
2     int i,j,temp;
3     for (i=1;i<len;i++){
4         temp = arr[i];
5         for (j=i;j>0 && arr[j-1]>temp;j--)
6             arr[j] = arr[j-1];
7         arr[j] = temp;
8     }
9 }
```


桶排序

将数组分到有限数量的桶里。每个桶再个别排序
(有可能再使用别的排序算法或是以递归方式继续
使用桶排序进行排序)

- 有n个数，每个数都是小于等于100的，只需要统计每种数字各出现了几次最后扫描一遍值域就可以排好序。

基数排序

- 从低位到高位排序，根据当前位的数值分到0~9 这10个桶里面，每一轮结束重新合并。
- 第step轮结束后，所有数相对于后step位来说是有顺序的。

```
int tong[10][N];
int top[10];
void radix_sort(int *a, int n) {
    int num = 1;
    for (int step = 0; step < 7; step++) {
        memset(top, 0, sizeof(top));
        for (int i = 0; i < n; i++) {
            int w = a[i] / num % 10;
            tong[w][top[w]++] = a[i];
        }
        int tot = 0;
        for (int i = 0; i < 10; i++) {
            for (int j = 0; j < top[i]; j++) {
                a[tot++] = tong[i][j];
            }
        }
        num = num * 10 ;
    }
}
```

归并排序

- 将待排序区间分成两半，先排左边一半，再排右边一半，最后合并

快速排序

- 对于当前要排序的区间，选取一个基准位置 p ，将小于 $a[p]$ 的数字放到左边，大于 $a[p]$ 的数字放到右边，然后递归排序。