

离散化

sdsy_N

- 给你一个序列，数值的范围比较大，值域可能是 $[0, 10^9]$ ，但是里面的数的个数是很少的，个数可能是 $[0, 10^5]$ ，有些时候我们需要利用这些值当作数组的下标来做，但是我们是不可能去开一个 10^9 大的数组的，因此，我们就要将这个序列里面的数映射到一些比较小的数。
- 离散化本质上可以看成是一种哈希

- 我们并不需要知道这些数据具体是多少，只需要知道他们的相对大小关系就可以了。
- 例如：
 - 5 1 1000000000 99999 3
 - 映射成
 - 3 1 5 4 2
- 这个过程就是离散化。
- 如何完成这个过程呢？

前置知识

- unique 函数
- lower_bound 函数 和 upper_bound 函数

- 排序
 - 去重(一般需要去重)
 - 映射(重新分配)
-
- 例如: $a[1..9] = \{ 6, 8, 4, 9, 5, 6, 7, 4, 6 \}$
 - 首先排序(sort)后得到 $b[1..9] = \{ 4, 4, 5, 6, 6, 6, 7, 8, 9 \}$
 - 去重(unique)后得到 $b[1..6] = \{ 4, 5, 6, 7, 8, 9 \}$
 - 最后在 b 中二分查找(lower_bound)原序列中的数的位置, 原序列就映射成了 $a[1..9] = \{ 3, 5, 1, 6, 2, 3, 4, 1, 3 \}$
 - 想要知道原序列中比如原来的第 i 个数是多少: $b[a[i]]$

离散化模板1

- 去重

```
cin>>n;
for(int i=1; i<=n; i++)
{
    cin>>a[i];
    b[i]=a[i];
}
sort(b+1, b+n+1);
m = unique(b+1, b+n+1) - b - 1;
for(int i=1; i<=n; i++)
    a[i] = lower_bound(b+1, b+m+1, a[i]) - b;
```

离散化模板2

- 去重

```
scanf("%d", &n);
for(int i = 1; i <= n; i++) scanf("%d", &a[i]);
for(int i = 1; i <= n; i++) {t[i].x = a[i]; t[i].id = i;}
sort(t + 1, t + n + 1); //排序
for(int i = 1; i <= n; i++)
{
    if(i == 1 || t[i].x != t[i-1].x) now++; //如果是和前面不同的数, now++ (因为已经排好序了)
    b[now] = t[i].x; a[t[i].id] = now; //此时的a[i]存的就是原来的a[i]是第几小的数
}
for(int i = 1; i <= n; ++i) printf("%d ", a[i]);
printf("\n");
for(int i = 1; i <= now; ++i) printf("%d ", b[i]);
printf("\n");
```

其他离散化方法：

- 借助结构体，不去重

```
cin>>n;
for(int i=1;i<=n;i++)
    scanf("%d",&a[i].x), a[i].id=i, b[i]=a[i].x;

sort(a+1,a+n+1,cmp);//手写 cmp, 从小到大
for(int i=1;i<=n;i++)
    b[a[i].id]=i;//映射
```


其他离散化方法：

- 借助 STL 的 map、set 等

课下任务：

- 自行研究 set 和 unordered_set 、 map 和 unordered_map 的异同
- 熟练使用这些容器

#1381. 区间和

#1384. [CF670C] Cinema 电影

题单

- #1385. [USACO3.1] Shaping Regions 形成的区域
- #1387. [BZOJ1206] 虚拟内存
- #941. [SCOI 2009] 生日礼物
- #901. [NOI2015] 程序自动分析
- #1392. [SDOI2009] 虔诚的墓主人
- #1393. [SHOI2007] 园丁的烦恼