1. C++的STL中unordered\_map使用hashtable实现的。查找的速度很快。但是遍历较慢。

Map使用红黑树实现的，查找的速度是o(logn)，但是遍历的速度很快（中序遍历）。

1. 虚函数
2. 初始化指针一定要赋值（比如NULL），否则会报错。
3. 优先队列与大根堆小跟堆
   1. 建堆复杂度为o(n),思考方式是，从倒数第二排节点开始下沉，每个节点最多下沉1次，从倒数第三排节点开始下沉，每个节点最多下沉2次，…这样算起来就是o(n)
   2. priority\_queue<node, vector<node>,cmp>：

第一个表示数据类型，第二个表示容器类型（一般是vector），第三个是自定义比较大小的函数。

<https://blog.csdn.net/AAMahone/article/details/82787184>

最近学习STL，发现STL默认都是使用()比较的，默认比较使用less（即'<'运算符），如sort(a,a+n)，默认将数组按照递增的顺序来排序（前面的元素<后面的嘛），但是优先队列的源码比较奇特，虽然按道理使用less比较应该默认是小根堆（即堆顶元素最小），但是priority\_queue<>默认是大根堆的，这是因为优先队列队首指向最后，队尾指向最前面的缘故！每次入队元素进去经排序调整后，优先级最大的元素排在最前面，也就是队尾指向的位置，这时候队首指向优先级最小的元素！所以虽然使用less但其实相当于greater，我们重载运算符的时候比较函数里面写>就相当于<排序方式，这点需要花点时间想想，再来说一说优先队列的这个类型，其实有三个参数：priority\_queue<class Type,class Container,class Compare>，即类型，容器和比较器，后两个参数可以缺省，这样默认的容器就是vector，比较方法是less，也就是默认大根堆，可以自定义写比较方法，但此时若有比较方法参数，则容器参数不可省略！priority\_queue<>的可支持的容器必须是用数组实现的容器，如vector，deque，但不能是list（推荐vector），比较方法可以写结构体重载()运算符，也可以用less，greater这些语言实现了的，但是灵活性不够，建议手写重载结构体，或者