参考文献:C++的pb_ds库在OI中的应用

__gnu_pbds::priority_queue 可合并堆

- 头文件 ext/pb_ds/priority_queue.hpp
  __gnu_pbds::priority_queue<T,greater,TAG>
- 函数：size(),empty(),push(T),top(),pop(),clear()
- 新增功能
  - begin(),end()获取iterator遍历
  - increase_key,decrease_key
  - 删除单个元素 erase(point_iterator)
  - point_iterator push(T)
  - 修改元素 modify(point_iterator,T)
  - 合并堆：q1.join(q2) 将q2合并到q1，q2被清空
- TAG:
  五种操作:push,pop,modify,erase,join
  - pairing_heap_tag(配对堆)：push,join$o(1)$，其余均摊$o(logn)$ (默认)
  - binary_heap_tag(二叉堆): 只支持push,pop 均摊$o(logn)$
  - binomial_heap_tag(二项堆): push均摊$o(1)$,其余$o(logn)$
  - rc_binomial_heap_tag: push$o(1)$，其余$o(logn)$
  - thin_heap_tag(斐波那契堆): push$o(1)$,不支持join,其余$o(logn)$,只有increase_key的话modity$o(1)$
- 合并,dij均使用: pairing_heap_tag
- 只有push,pop,join: binary_heap_tag

__gnu_pbds::tree

- 头文件 ext/pb_ds/assoc_container.hpp
  ext/pb_ds/tree_policy.hpp
  __gnu_gbds::tree<key,T,TAG,Node_Update>
- 函数类似于map：begin(),end(),size(),empty(),clear(),
  find(key),lower_bound(key),upper_bound(key),
  erase(iterator),erase(key),insert(<key,T>),operator
- 第二个参数改为null_type(null_mapped_type)即为set
- TAG:
  - rb_tree_tag
  - splay_tree_tag
- 寻找第order+1小的元素,order过大返回end()：iterator find_by_order(order)
- 询问有多少个比key小的元素：order_of_key(key)
- t1.join(t2)将t2所有元素移动到t1,t1、t2值域不能相交
- t1.split(key,t2)清空t2，把所有大于key的元素移动到other
- 自带的Node_Update:tree_order_statistics_node_update统计子树大小

- 自定义Node_Update

```
template<class Node_CItr,class Node_Itr,class Cmp_Fn,class _Alloc>
struct my_node_update{
    virtual Node_CItr node_begin() const =0;
    virtual Node_CItr node_end() const =0;
    typedef int metadata_typde;//节点记录的额外信息的类型
}
/*
将系欸但it的信息更新为其左右孩子的信息
传入end_it表示空节点
*/
inline void operator()(Node_Itr it,Node_CItr end_it)
{
    Node_Itr l=it.get_l_child(),r=it.get_r_child();
    int left=0,right=0;
    if(l!=end_it) left=l.get_metadata();
    if(r!=end_it) right=r.get_metadata();
    const_cast<metadata_type &>(it.get_metadata())
     =left+right+(*it)->second;
}
inline int prefix_sum(int x)
{
    int ans=0;
    Node_CItr it=node_begin();
    while(it!=node_end())
    {
        Node_CItr l= it.get_l_child(),r=it.get_r_child();
        if(Cmp_Fn()(x,(*it)->first)) it=l;
        else{
            ans+=(*it)->second;
            if(l!=node_end())
               ans+=l.get_metadata();
               it=r;
        }
    }
    return ans;
}
inline int interval_sum(int l,int r)
{
    return prefix_sum(r)-prefix_sum(l-1);
}
```

get_l_child,get_r_child获取左右孩子,(**it)获取节点信息，get_metadata获取节点额外信息

hash_table

- 头文件 ext/pb_ds_assoc_container.hpp
  ext/pb_ds/hash_policy.hpp
  __gnu_pbds::cc_hash_table<key,mapped>(拉链法)
  __gun_pbds::gp_hash_table<key,mpped>（查探法较快）