PBDS - Policy Based DSA/Order Set

প্রথম কথা হলো PBDS খুবই সহজ Data structure.

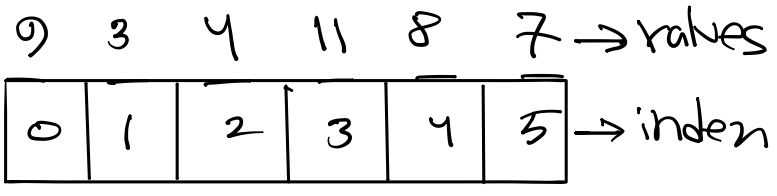
PBDS — set/multiset এর সকল কাজই করতে পারে। (মিথ্যে কথা, multiset থেকে erase করতে পারে না)

Ami set/multiset pari, tahole keno shikbo?

দুইটা কারণে শিখব।

i. order_of_key(data) : data এর চেয়ে strictly less কয়টি Value আছে সেটি O(log N) এ কাউন্ট করতে পারব। নর্মালি করতে গেলে O(.....

ii. find_by_order(data) : 0-based index চিন্তা করে একটা ইনডেক্স দিয়ে ঐ ইনডেক্সে কোন value আছে সেটি O(log N) এ বের করা যায়। হুহ...



p.find_by_order(3) = 11

Implementation of PBDS

```cpp
#include <bits/stdc++.h>

#include<ext/pb_ds/assoc_container.hpp>
#include<ext/pb_ds/tree_policy.hpp>
using namespace __gnu_pbds;

using namespace std;

template <typename T> using pbds = tree<T, null_type, less<T>, rb_tree_tag, tree_order_statistics_node_update>;

int main()
{

   return 0;
}
```

মনমতো নাম দেয়া যাবে।

set

⊛ less এর স্থানে `less_equal` দিলে জিনিসটা `multiset` এর মতোন কাজ করবে।

⊛ অনেক সময় multiset রাখা লাগে, আবার সেগুলো erase করার প্রয়োজন পড়ে। এজন্য

```
pbds < pair<int, int> > p;
p.insert({data, i});
```

রাখলে বহুত ফায়দা হবে। :D

তখন value, index খুঁজে পাবো তো?

হ্যাঁ, `map <int, int> mp;` ব্যবহার করা যেতে পারে। Your wish...