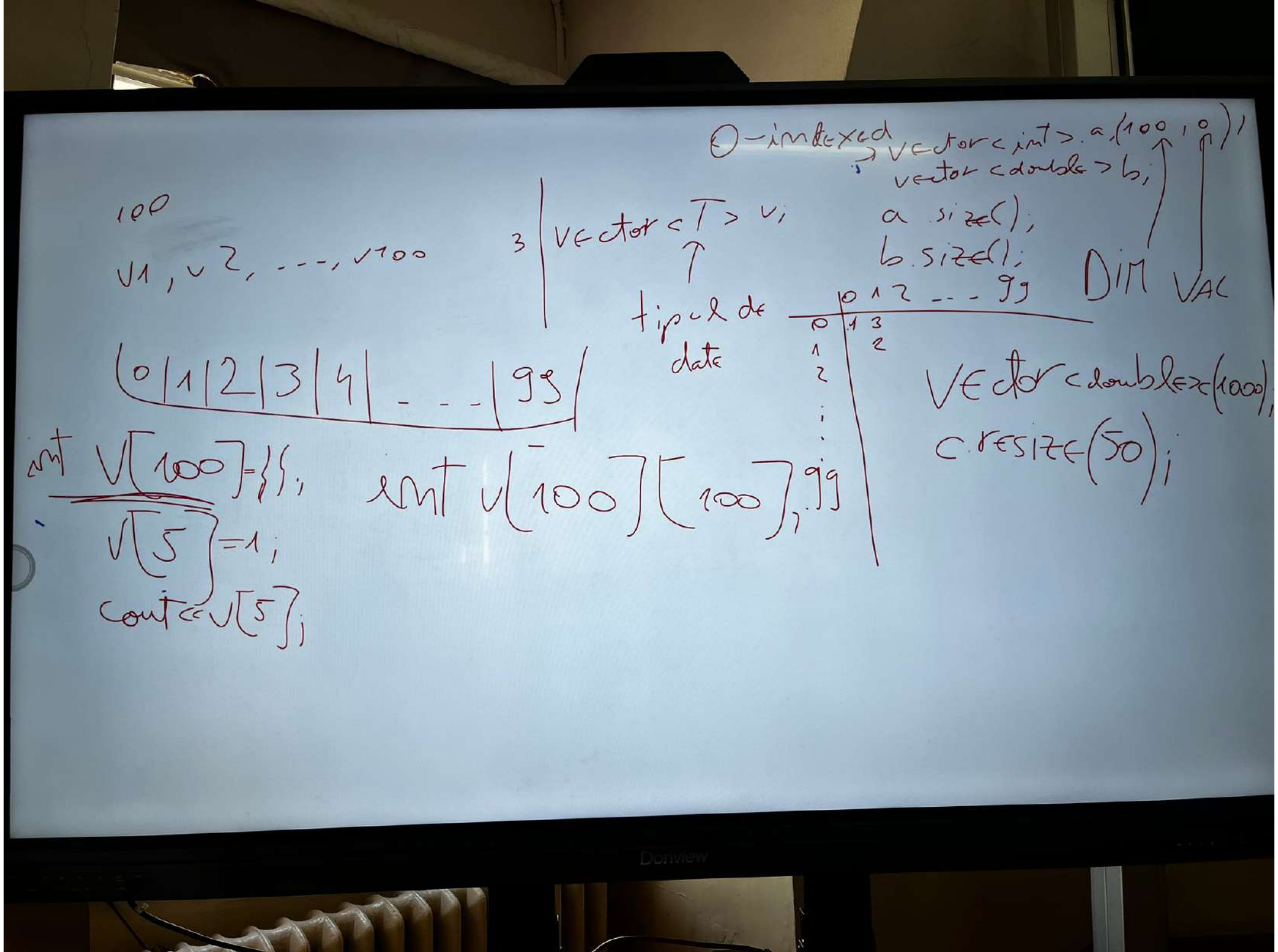
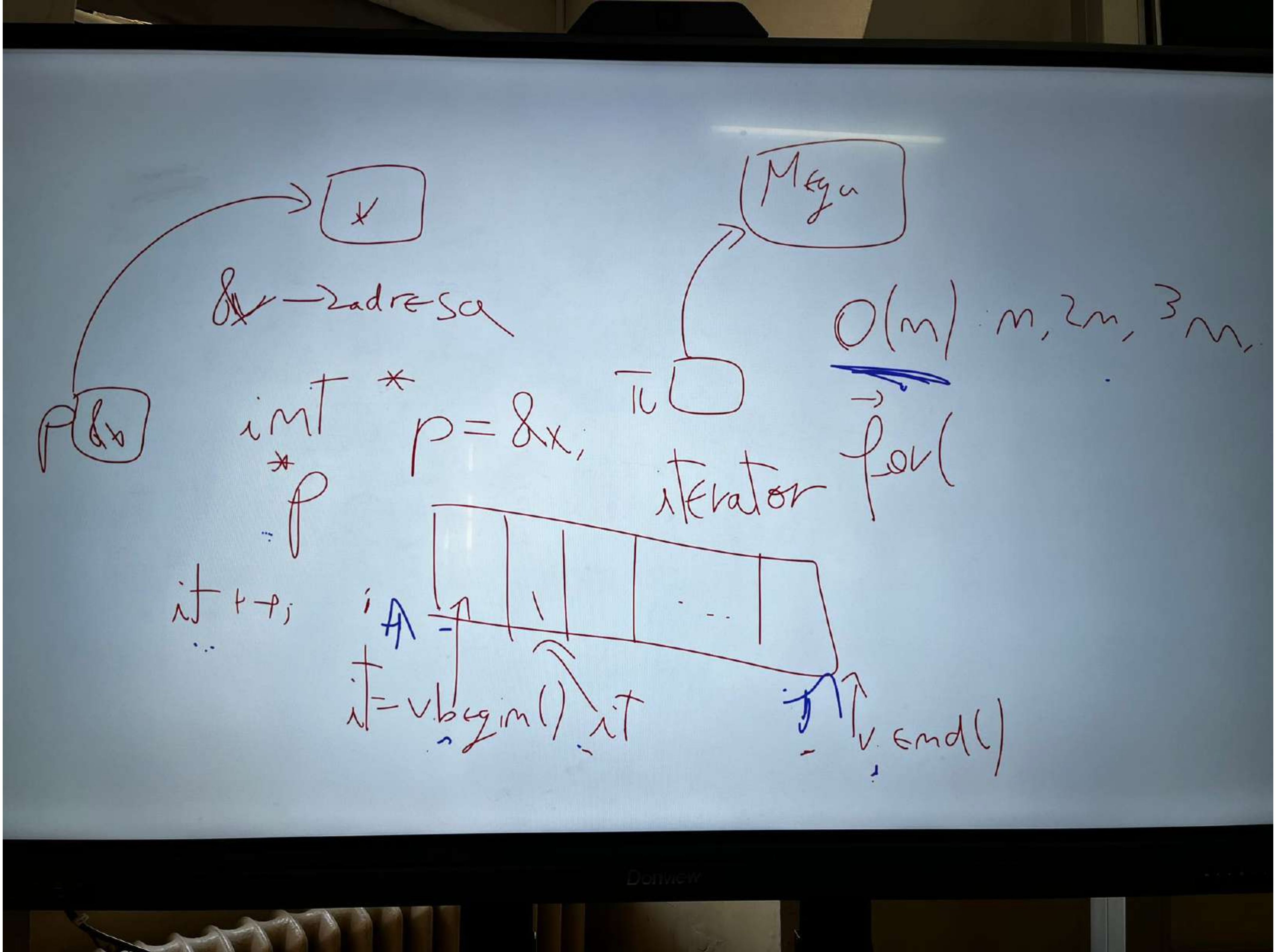
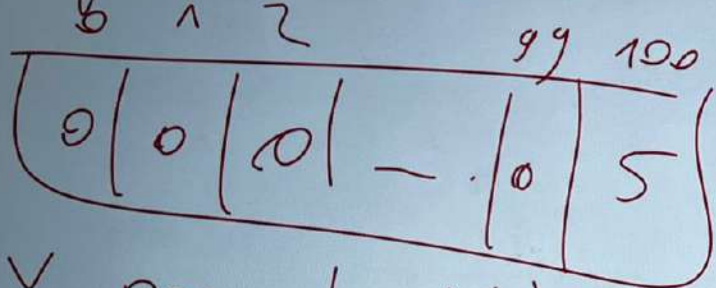

0-indexed
vector<int> a(100,0);
vector<double> b;
a.size();
b.size();
DIM
VAC
100
v1, v2, ---, v100
3
vector<T> v;
tipul de date
0 1 2 ... 99
vector<double>c(1000);
c.resize(50);
int v[100]={};
int v[100][100];
v[5]=1;
cout<<v[5];

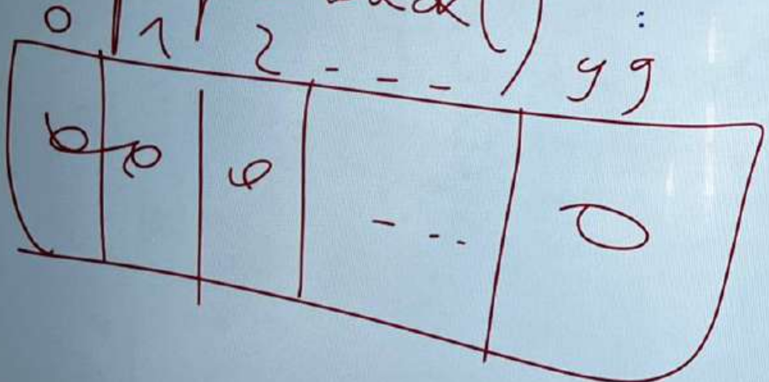

x
&x → adresa
p
&x
int * p = &x;
*p
it++;
iterator
for(
O(n) n, 2n, 3n
it = v.begin()
it
v.end()

vector<int> v(100, 0).

v.push_back(5);



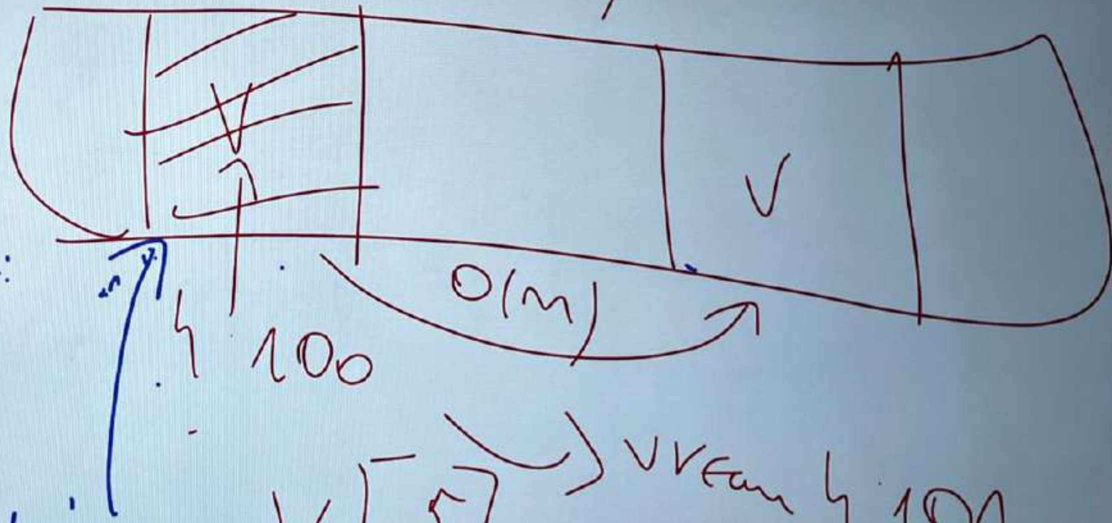
v.pop_back()



1 + 2 + 1 + ... + 1 + 400 + ... + 5

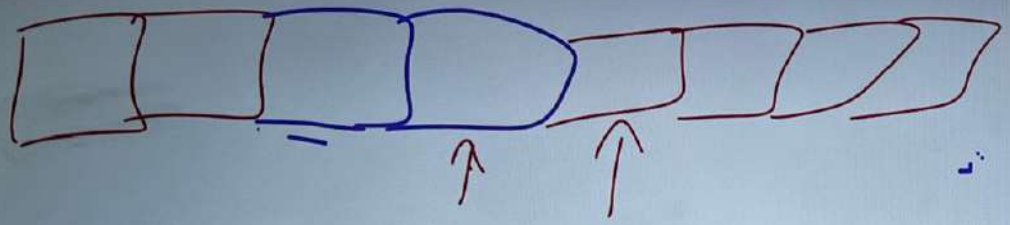
O(1) amortizat

O(n) vs O(1)



v[5] → vrem la 101

v + 5 · 4



int v[100];

sort(v, v+n, cmp);

sort(v.begin(), v.end());

1 → 2 → 4 → 8 → 16 → 32 → 64

1 2 4 8 16 32

100000 → 200000

100000 + 1 · 99 999

100000 < 2

O(1) amortizat

pg-back() → a lot

50000 el. vărma

resize la 100 000

char c[100];

vector<char> v;

String s;

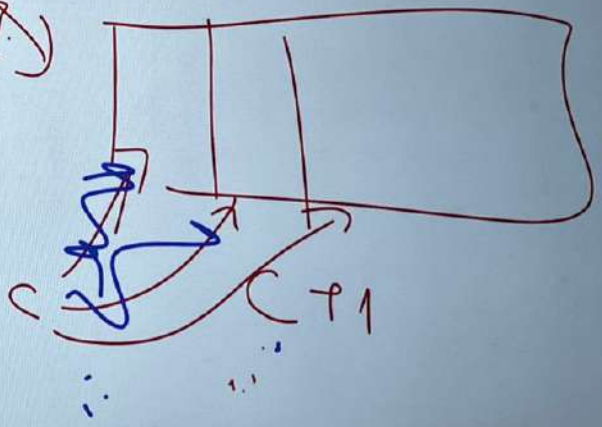
char* c = new char[100];

s = "xyz";

t = "abc";

for(int i=0; i<(int)t.size(); i++)

s.push_back(t[i]);



s = s + "abc"; → O(n)

s += "abc" → O(n)

vector<bool> v;

bitset<DIM> a;

const int NMAX = 100000;

bitset<NMAX> a;

bitset<100000> a,b; vs 800 000

onst int v=5; const

$(\frac{N}{64})$

a f = b;

bool 0 1 2 3 ... 7

bool b[100];

ll1 ll2 ll3 ll4

$\frac{100000}{64} = 1563$

$\frac{256}{64} = 4$ long longuri

Tuple< T1, T2, T3, ... >   vector<pair< T1, T2 >> v;

p.first   p.second

100000

B → 8

KB → 1024

MB → 2024

vector<int>   vector<vector<int>> v;

map< T1, T2 > m;

key   value

vector<string> v;
v[1] = "Maria ... "

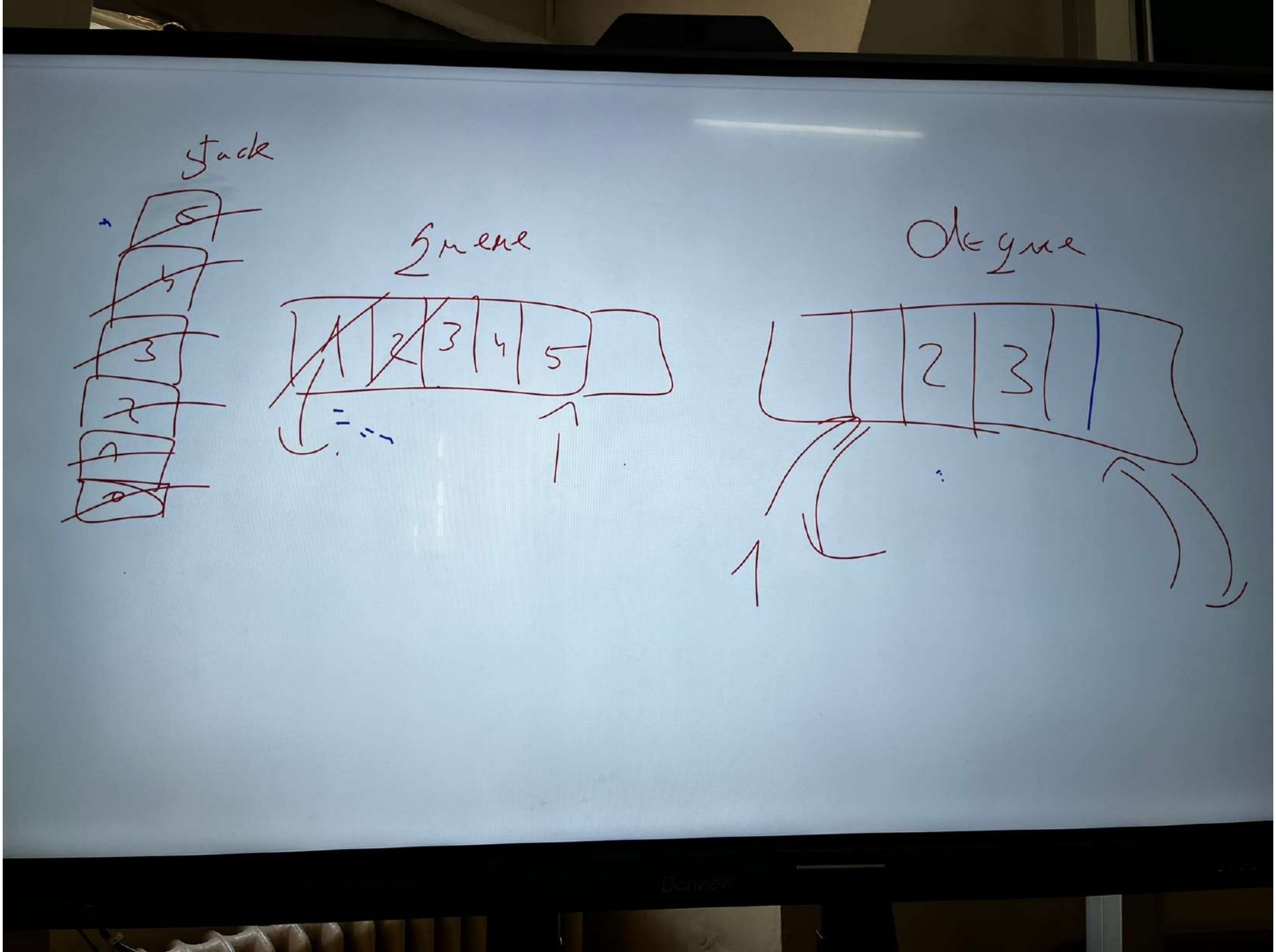
Red Black Trees

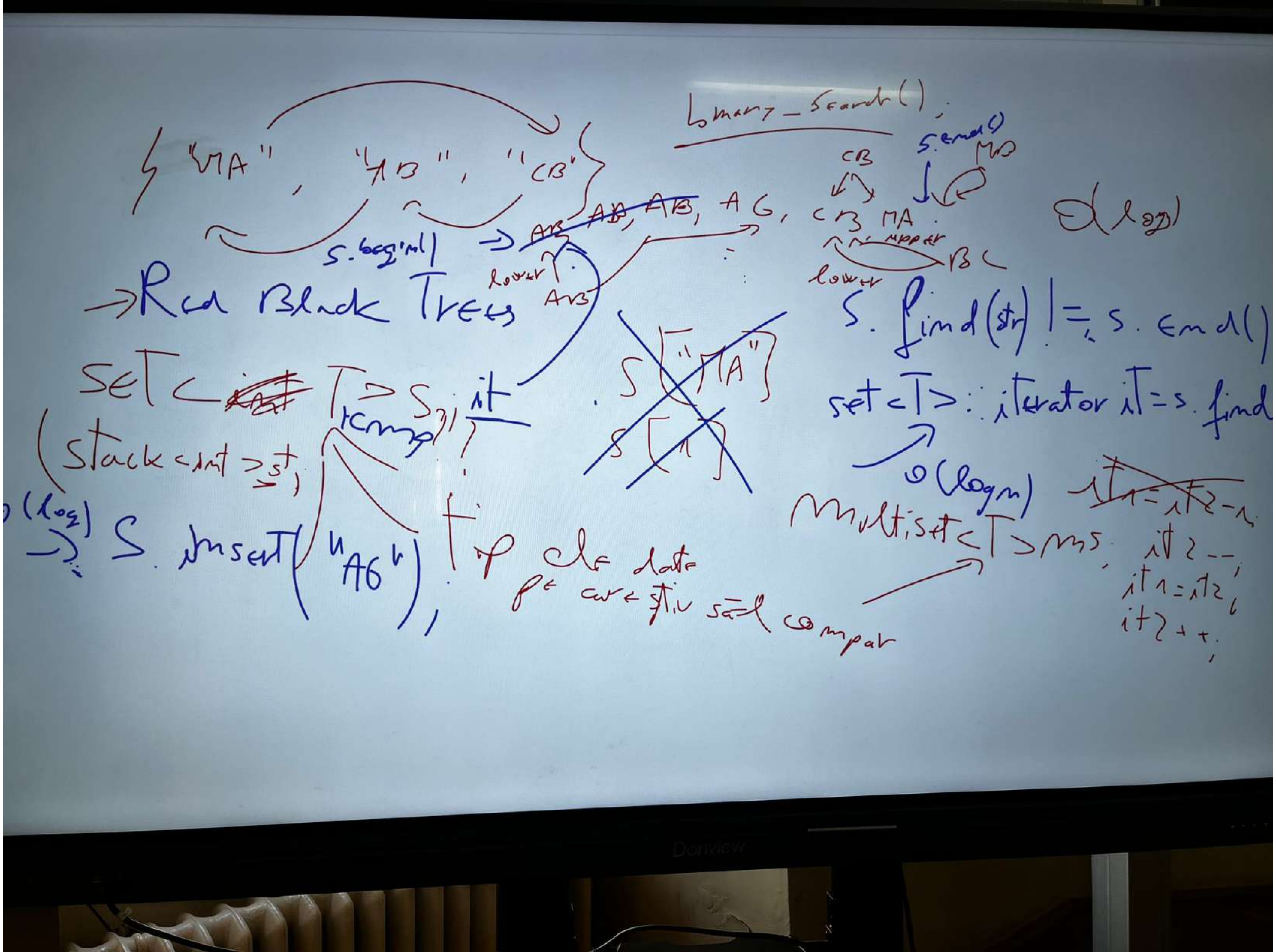
insert
range
find
→ O(log n) → m["Maria Alima"]++;

map< string, int > m;   (*it).first

m.insert(...)

m["5"] = 7;   m["5"] = 9   it->first


stack
Queue
Dequeue



{"MA", "AB", "CB"}
binary_search();
s.end()
s.begin()
lower
upper
Red Black Trees
set<T> s;
stack<int> st;
s.insert("AG");
s.find(str) != s.end()
set<T>::iterator it = s.find
O(logn)
Multiset<T> ms;