খুবই সহজ কিন্তু দরকারি একটা টেকনিক নিয়ে আলোচনা করবো আজকে। STL এর ম্যাপ ব্যবহার করে আমরা অ্যারে কম্প্রেশন করবো। মনে করো কোনো একটা প্রবলেমে তোমাকে বলা হলো একটি অ্যারেতে ১ লাখটা সংখ্যা দেয়া থাকবে যাদের মান হবে ০ থেকে সর্বোচ্চ ১০০০। এখন যেকোনো আমি একটি সংখ্যা বললে তোমাকে বলতে হবে অ্যারের কোন কোন পজিশনে সংখ্যাটি আছে। যেমন মনে করো ইনপুট অ্যারেটা হলো:

১ ০ ০ ২ ৫ ২ ১ ০ ৪ ৫ ১ ২

খুবই সহজ সলিউশন হলো একটি ভেক্টর নেয়া। i তম পজিশনে x সংখ্যাটি পেলে ভেক্টরের x তম পজিশনে পুশ করে রাখবে i। তাহলে ইনপুট নেবার পর ভেক্টরগুলোর চেহারা হবে:

[0]->1 2 7
[1]->0 6 10
[2]->3 5 11
[3]->empty
[4]->8
[5]->4 9

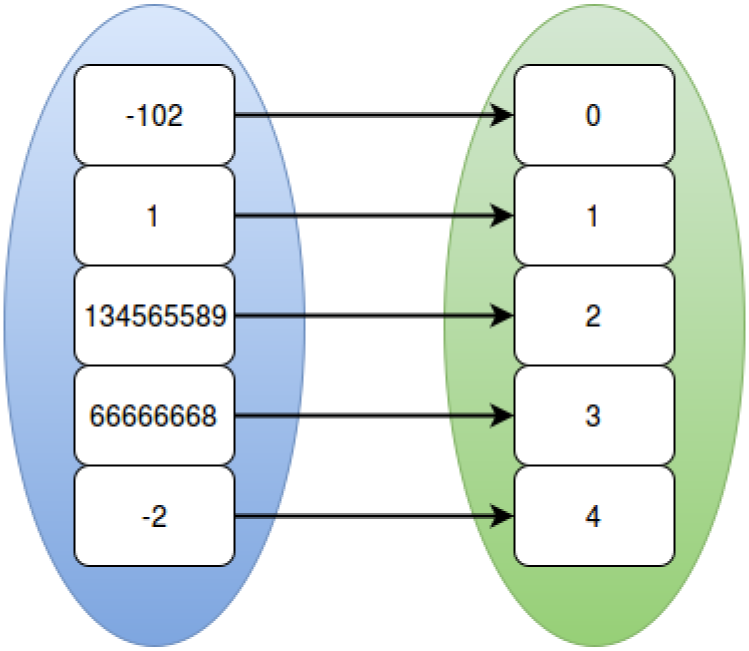
কোন সংখ্যা কোন কোন পজিশনে আছে তুমি পেয়ে গেলে। এখন যদি বলে 2 কোথায় কোথায় আছে তাহলে তুমি ২ নম্বর ইনডেক্সে লুপ চালিয়ে 3,5,11 প্রিন্ট করে দাও। 1001 সাইজের ২-ডি ভেক্টর দিয়েই কাজ হয়ে যাবে।

এখন তোমাকে বলা হলো সংখ্যাগুলো নেগেটিভও হতে পারে এবং সর্বোচ্চ 2^30 পর্যন্ত হতে পারে। এবার কি এই পদ্ধতি কাজ করবে? একটি সংখ্যা -100 বা 2^30 হলে তুমি সেটার পজিশনগুলা কোন ইনডেক্সে রাখবে? অবশ্যই তুমি এতো বড় ভেক্টর ডিক্লেয়ার করতে পারবেনা অথবা নেগেটিভ ইনডেক্স ব্যবহার করতে পারবেনা। ধরো ইনপুট এবার এরকম:

input[]={-102,1,134565589,134565589,-102,66666668,134565589,66666668,-102,1,-2}

একটা ব্যাপার লক্ষ্য করো, সংখ্যা দেয়া হবে সর্বোচ্চ 10^5 বা ১ লাখটা। তাহলে অ্যারেতে মোট ভিন্ন ভিন্ন সংখ্যা হতে পারে কয়টা? অবশ্যই সর্বোচ্চ ১ লাখটা। আমরা প্রতিটি ভিন্ন ভিন্ন সংখ্যাকে ছোটো কিছু সংখ্যার সাথে one-to-one ম্যাপিং করে ফেলবো। উপরের অ্যারেতে ভিন্ন ভিন্ন সংখ্যা গুলো হলো:

-102,1,134565589,66666668,-2



দুইটা উপায় আছে ম্যাপিং করার। একটা হলো STL এর map ব্যবহার করে। map এর কাজ হলো যেকোনো একটি সংখ্যা, স্ট্রিং বা অবজেক্টকে অন্য একটি মান অ্যাসাইন করা যেটা আমরা উপরের ছবিতে করেছি। আমাদের এই প্রবলেমে কোন সংখ্যা কার থেকে বড় বা ছোটো সেটা দরকার নাই তাই সর্ট না করেই ম্যাপিং করে দিতে পারি। নিচের কোডটা দেখো:

```
void compress() {
 map < int, int > mymap;
 int input[] = {
  -102,
  1,
  134565589,
  134565589,
  -102,
  66666668,
  134565589,
  66666668,
  -102,
  1,
  -2
 };
 int assign = 0, compressed[100], c = 0, n = sizeof(input) / sizeof(int); //array size;
 for (int i = 0; i < n; i++) {
  int x = input[i];
  if (mymap.find(x) == mymap.end()) { //x not yet compressed
   mymap[x] = assign;
   printf("Mapping %d with %d\n", x, assign);
   assign++;
```

```
    }
    x = mymap[x];
    compressed[c++] = x;
  }
  printf("Compressed array: ");
  for (int i = 0; i < n; i++) printf("%d ", compressed[i]);
  puts("");
}
```

কোডের আউটপুট হবে:

```
Mapping -102 with 0
Mapping 1 with 1
Mapping 134565589 with 2
Mapping 66666668 with 3
Mapping -2 with 4
Compressed array: 0 1 2 2 0 3 2 3 0 1 4
```

আমরা ইনপুট অ্যারেতে যখনই একটু নতুন সংখ্যা পাচ্ছি সেটাকে একটা ভ্যালু অ্যাসাইন করে দিচ্ছি এবং আসল ভ্যালুকে ম্যাপের ভ্যালু দিয়ে রিপ্লেস করে আরেকটি অ্যারেতে রেখে দিয়েছি। এরপরে আমরা নতুন অ্যারেটা ব্যবহার করে প্রবলেমটা সলভ করতে পারবো। কুয়েরিতে যদি বলে -2 কোথায় কোথায় আছে বের করো তাহলে তুমি আসলে 4 কোথায় কোথায় আছে বের করবে কারণ mymap[-2]=4।

অনেক সময় কমপ্রেস করার পরেও কোন সংখ্যাটি কার থেকে বড় এটা দরকার হয়। যেমন তোমাকে বলতে পারে যে একটি ভ্যালু নেয়ার পর ছোটো কোনো ভ্যালু আর নিতে পারবেনা। এই তথ্যটা কিভাবে রাখবে উপরের পদ্ধতিতে ১ এর থেকে -২ এ অ্যাসাইন করা ভ্যালু বড় হয়ে গিয়েছে। যদি তুমি সেটা না চাও তাহলে আগে ভিন্ন ভিন্ন ভ্যালুগুলো আরেকটা অ্যারেতে সর্ট করে নাও।

sorted[]={-102,-2,1,66666668,134565589}

এইবার sorted অ্যারেটা ব্যবহার করে ম্যাপিং করে ফেলো। এ ক্ষেত্রে আসলে যে সংখ্যাটি sorted অ্যারেতে যে পজিশনে আছে সেটাই হবে তার ম্যাপ করা ভ্যালু।
এটা যদি বুঝতে পারো তাহলে নিশ্চয়ই ম্যাপ ছাড়াই বাইনারি সার্চ করে তুমি কমপ্রেস করে ফেলতে পারবে অ্যারেটাকে। ইনপুট অ্যারের উপর লুপ চালাও, প্রতিটি x এর জন্য দেখো sorted অ্যারেতে x এর অবস্থান কোথায়। সেই মানটা তুমি আরেকটা অ্যারেতে রেখে দাও:

input[]={-102,1,134565589,134565589,-102,66666668,134565589,66666668,-102,1,-2}
compressed_input[]={0,2,4,4,0,3,4,3,0,2,1}

কুয়েরির সময়ও বাইনারী সার্চ করে কমপ্রেস করার মানটা বের করে কাজ করতে হবে।
তুমি ম্যাপ বা বাইনারি সার্চ যেভাবে ইচ্ছা কমপ্রেস করতে পারো। প্রতিবার map এক্সেস করতে logn কমপ্লেক্সিটি লাগে যেটা বাইনারী সার্চের সমান। stl এ বিভিন্ন class,ডাইনামিক মেমরির ব্যবহারের

জন্য map কিছুটা স্লো, তবে টাইম লিমিট খুব tight না হলে খুব একটা সমস্যা হবেনা। map ব্যবহার করলে কোডিং সহজ হয়।

কোনো কোনো গ্রাফ প্রবলেমে নোডগুলা আর এজগুলো কে string হিসাবে ইনপুট দেয়। যেমন ধরো গ্রাফের ৩টা এজ হলো:

3
BAN AUS
AUS SRI
SRI BAN

map এ string কেও integer দিয়ে ম্যাপিং করা যায়। এই সুবিধাটা ব্যবহার করে প্রতিটি নোডকে একটি ভ্যালু অ্যাসাইন করবো:

```
map < string, int > mymap;
int edge, assign = 0;
cin >> edge;
for (int i = 0; i < edge; i++) {
  char s1[100], s2[100];
  cin >> s1 >> s2;
  if (mymap.find(s1) == mymap.end()) {
    printf("Mapping %s with %d\n", s1, assign);
    mymap[s1] = assign++;
  }
  if (mymap.find(s2) == mymap.end()) {
    printf("Mapping %s with %d\n", s2, assign);
    mymap[s2] = assign++;
  }
  int u = mymap[s1];
  int v = mymap[s2];
  cout << "Edge: " << u << " " << v << endl;
}
```

অনেক সময় বলা হতে পারে নোডগুলোর মান হবে -2^30 থেকে 2^30 পর্যন্ত কিন্তু সর্বোচ্চ নোড হবে ১০০০০ টা, তখন আমরা নোডগুলোকে একই ভাবে ছোটো ভ্যালু দিয়ে ম্যাপিং করে ফেলবো।

2-d grid ও কমপ্রেস করে ফেলা যায় অনেকটা এভাবে। সেটা নিয়ে আরেকদিন আলোচনা করতে চেষ্টা করবো, তবে নিজে একটু চিন্তা করলেই বের করতে পারবে।

প্রবলেম:
Drunk
Babel
A Node Too Far