

Tipos de datos simples

- numéricos: int, long long, float, double
- alfabéticos: char, string
- arreglos: “listas” de los anteriores

Pair

Que son?

Estructura de datos simple que nos permite crear un par de tipos a determinar.

Ejemplo

```
1 pair<int , int> A;  
2 pair<int , string> B;  
3 A = {1, 2};  
4 B = {3, "Hola mundo"};  
5 cout << A.first << " " << A.second << endl;  
6 cout << B.first << " " << B.second << endl;  
7
```

Problema

Enunciado

Nos dan un arreglo con n números.

Sea el "rango" de un arreglo = $\text{Max del arreglo} - \text{Min del arreglo}$.

Podemos eliminar solo elemento, y queremos minimizar el valor del rango.

Ejemplo

```
4
1 3 3 7
// Responder 2
2
1 100000
// Responder 0
5
14 11 1 12 10
// Responder 4
```

Vector

Qué es un vector?

Estructura de datos que nos permite tener elementos en una memoria contigua que se genera dinámicamente en $O(1)$ amortizado

Ejemplo

```
1  int n, element; cin >> n;
2  vector<int> v;
3  for(int i = 0; i < n; i++) {
4      cin >> element; v.push_back(element);
5  }
6  for(int i = 0; i < v.size(); i++) {
7      cout << v[i] << " ";
8  } cout << endl;
9  // Otra forma de recorrer un vector
10 for(int x : v) {
11     cout << x << " ";
12 } cout << endl;
```

Solución al problema

```
int n; cin >> n;
vector<int> v;
for(int i = 0; i < n; i++) {
    int a; cin >> a;
    v.push_back(a);
}
sort(v.begin(), v.end());
int primero = v[n - 2] - v[0];
int segundo = v[n - 1] - v[1];
cout << min(primero, segundo) << endl;
```

Principales operaciones / métodos

- size
- Operador []
- front
- back
- push_back
- pop_back
- insert
- erase
- swap
- clear

Problema

Enunciado

Nos dan n palabras, y queremos saber cuantas veces se repite cada una.

Ejemplo

```
10 // 10 palabras
casa vaca ababa casa vaca casa asca ababa casa acasa
// Resultado:
// Casa 4
// vaca 2
// ababa 2
// asca 1
// acasa 1
```

Map

Qué es un map?

Son contenedores que guardan elementos, relacionados con un valor, en algún orden.

Ejemplo

```
1 int main() {  
2     int n, m; cin >> n >> m;  
3     map<string, string> dict;  
4     string es, en, w;  
5     for(int i = 0; i < n; i++) {  
6         cin >> es >> en; dictionary[es] = en;  
7     }  
8     vector<string> text;  
9     for(int i = 0; i < m; i++) { cin >> w; text.pb(w); }  
10    for(string word : text) {  
11        cout << dictionary[word] << endl;  
12    }  
}
```


Solución al problema

```
1  int n; cin >> n;
   map<string , int> m;
3  for(int i = 0; i < n; i++) {
       string word; cin >> word;
5       m[word]++;
   }
7  for(pair<string , int> x : m) {
       cout << x.first << " " << x.second << endl;
9  }
```

Principales operaciones / métodos

- size
- operador []
- insert
- erase
- swap
- find
- lower_bound
- upper_bound
- clear

Problema

Enunciado

Tenemos un arreglo con n números. Nos llegan q preguntas donde nos pasan un número x . Debemos responder con el valor mas cercano y mayor a x .

Ejemplo

```
1 5 5 // 5 elementos , 5 preguntas
2 8 4 5 9 10 // Arreglo
3 9 // Responder 10
4 7 // Responder 8
5 6 // Responder 8
6 20 // Responder -1
7 1 // Responder 4
```

Set

Qué es un set?

Son contenedores que guardan elementos no repetidos en un orden específico (ordenados bajo algún criterio).

Ejemplo

```
1  int n, element; cin >> n;
   set<int> s;
3  for(int i = 0; i < n; i++) {
   cin >> element; s.insert(element);
5  }
   itlow = s.lower_bound(30);
7   itup = s.upper_bound(60);
   s.erase(itlow, itup);
9
```

Solución al problema

```
1 int n, q, element; cin >> n >> q;
2 set<int> s;
3
4 for(int i = 0; i < n; i++) {
5     cin >> element; s.insert(element);
6 }
7
8 for(int i = 0; i < q; i++) {
9     cin >> element;
10    auto it = s.upper_bound(element);
11    if(it == s.end()) cout << -1 << endl;
12    else cout << *it << endl;
13 }
```

Principales operaciones / métodos

- size
- insert
- erase
- swap
- find
- lower_bound
- upper_bound
- clear

Templates

Para que son?

Nos permite escribir código genérico que puede ser usado con varios tipos de datos. Nos permiten parametrizar estas clases para adaptarlas a cualquier tipo de dato

De que manera?

Poniendo el tipo entre <>

Ejemplo

```
1  vector<int> A;  
   vector<vector<pair<int, string>>> K;  
3  queue<vector<int>> Q;  
   queue<struct nodo> bfsQ;
```