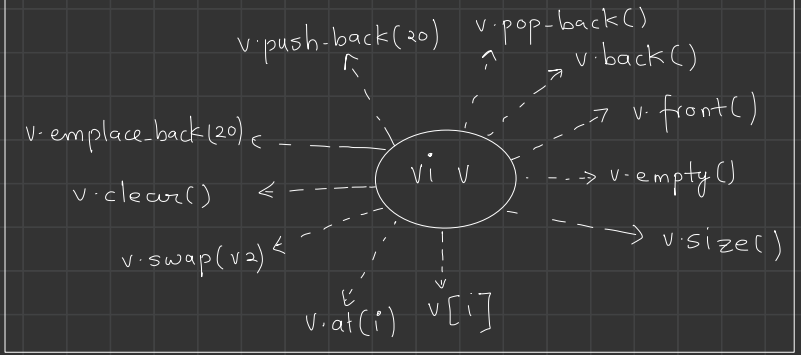
① { Vectors }

```
# define   vi   vector<int>
```



size — element to be filled

```
vi  v(5,-1);
// -1 -1 -1 -1 -1
```

```
vi  w(5);
```
↳ vector of size 5

```
vector<vector<int>>  v(5, vector<int>(6,-1));
```
↳ creates a matrix of size 5×6 with all elements −1

```
for(auto& x : v)   // can use const auto& x : v as well
{
    cout << x << " ";
}
cout << endl
```

```
vi  v{1,2,3,4,5};
```

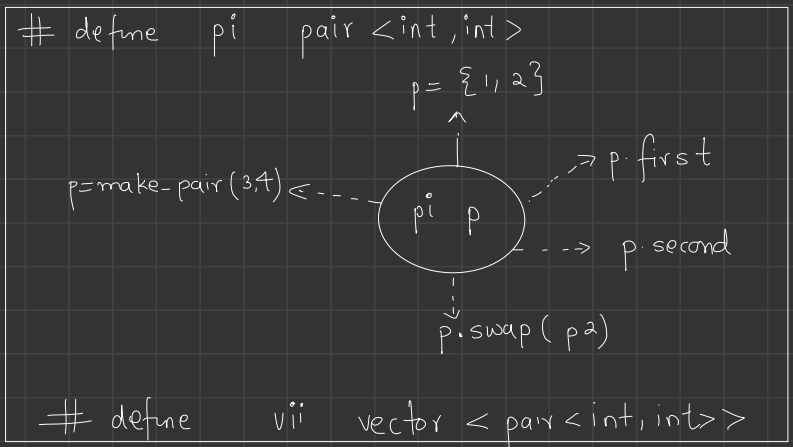
1 2 3 4 5

v.begin() ↑ — v.end() ↑

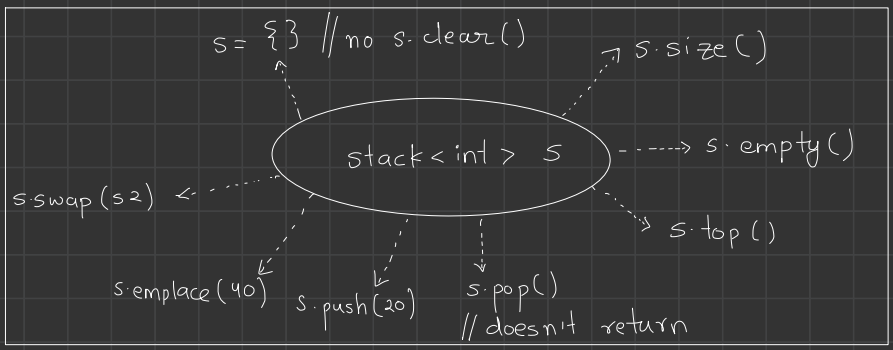
```
sort(v.begin(), v.end())   ⟶   sort(all(v))

# define  all(v)  v.begin(), v.end()
```

② { pair }



③ { stack }

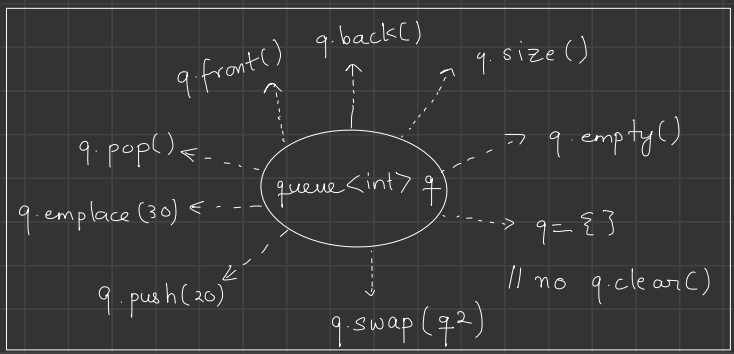


// Range based for loops and iterative for loops are forbidden they don't have `iterators` (s.begin() & s.end() ✗)

```
for ( auto & x : s)
    cout << x << " ";
cout << endl;
```

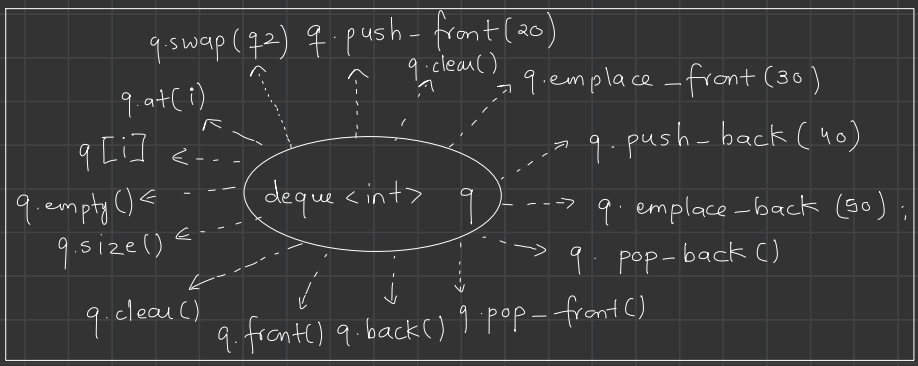
→ This won't work

④ { queue }



// Range based for loops and iterative loops are forbidden to access queue elements. So q.begin() and q.end() are not applicable.

⑤ { deque } // Doubly ended queue



// iterators also exist q.begin() and q.end()

```
for (int i=0 ; i<10 ; i++)
{
    cout << q[i] << " ";
}
```

```
for (auto& x : q)
{ cout << x << " ";
}
cout << endl;
```

```
deque<int>::iterator it = q.begin();

while (it != q.end())
{
    cout << *it << " ";
}
```