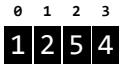
Growing & Shrinking

Unlike the built-in array type, the size of a `vector` is not fixed; it can grow or shrink at runtime. The `push_back()` member function appends a new element **to the end** of the `vector`. If the `vector` is full, **it is expanded**. Here's an example:

```
vector<int> v;      // size is 0
v.push_back(1);     // size is now 1
v.push_back(2);     // size is now 2
v.push_back(5);     // size is now 3
```

If `v` is an empty `vector<int>`, executing the code above adds these three elements to the end of `v`. Afterwards, `v` looks like the illustration here.

You can add additional elements at any time. Later, for example, you could call `v.push_back(4);` which would add the value `4` to the end of the `vector`, like this.



The `pop_back()` member function removes the element at **the end** of the `vector` and **decreases** its size. If the `vector` is empty, calling `pop_back()` is undefined behavior.After calling `v.pop_back()` on the `vector`, its contents and size are back where it started.