

```
void push_back(const T& data);
- verifica se cheio (full)
- contents[size_] = data
- size_++

void push_front(const T& data);
- verifica se cheio (full)
- for (i = size_; i > 0; i--) {
    contents[i] = contents[i-1];
}
- contents[0] = data
- size_++

void insert(const T& data, std::size_t index);
- verifica se cheio (full)
- o 'index' deve estar entre 0 e size()
- for (i = size_; i > index; i--) {
    contents[i] = contents[i-1];
}
- contents[index] = data
- size_++

void insert_sorted(const T& data);
- verifica se cheio (full)
- i = 0
- while (i < size_ && contents[i] < data) {
    i++;
}
- insert(data, i);

T pop(std::size_t index);
- verifica se vazia (empty)
- aux = contents[index]
- for (i = index + 1; i < size_; i++) {
    contents[i-1] = contents[i]
}
- size_--
- returna aux

T pop_back();
- verifica se vazia (empty)
- size_--
- returna contents[size_]

T pop_front();
- verifica se vazia (empty)
- aux = contents[0]
- for (i = 1; i < size_; i++) {
    contents[i-1] = contents[i]
}
- size_--
- returna aux

void remove(const T& data);
- verifica se vazia (empty)
- i = 0
- while (i < size_ && contents[i] != data) {
    i++;
}
- if (i < size_) {
    pop(i)
}
```