ContainerAdapters

January 18, 2026

1 Container Adapters: Stack, Queue, Priority Queue

1.1 External Resources

- C++ Reference for Container Adapters: https://en.cppreference.com/w/cpp/container.html
- YouTube Podcast - https://youtu.be/_8LsIm9rwk0
- YouTube Video - https://youtu.be/zAdiE1fPr00
- NotebookLM learning materials - https://notebooklm.google.com/notebook/f1bcefe5-531e-4606-80eb-9e4bc1059fe0

1.2 Overview

- Container adapters are special types of containers in C++ that provide a specific interface for data storage and retrieval, built on top of other standard containers.
- The most commonly used container adapters are `stack`, `queue`, and `priority_queue`. These adapters restrict the way elements can be accessed and modified, providing a more controlled interface.
- Here are brief descriptions of each:

1.3 Stack

- A stack is a Last In First Out (LIFO) data structure.
- Elements can only be added or removed from the top of the stack.
- Common operations: `push()`, `pop()`, `top()`, `empty()`, `size()`
- Example usage:

[1]:
```
#include <iostream>
#include <stack>
using namespace std;
```

[3]:
```
stack<int> num_stack;
num_stack.push(10);
num_stack.push(20);
num_stack.push(30);
num_stack.push(1);
num_stack.push(5);
```

[4]:
```
num_stack
```

[4]: @0x7fa4d67ec080

[5]:
```
cout << num_stack.top() << endl;
```

```
5
```

[ ]:
```
num_stack.size();
```

[ ]: 5

[7]:
```
num_stack.pop();
```

[8]:
```
num_stack.empty()
```

[8]: false

[10]:
```
num_stack.top()
```

[10]: 1

[11]:
```
// accessing all elements in the stack
// Note: must pop elements to access them
while (!num_stack.empty())
{
    cout << num_stack.top() << endl;
    num_stack.pop();
}
```

```
1
30
20
10
```

1.4 Queue

- A queue is a First In First Out (FIFO) data structure.
- Elements are added at the back and removed from the front.
- Common operations: `push()`, `pop()`, `front()`, `back()`, `empty()`, `size()`
- Example usage:

[13]:
```
#include <iostream>
#include <string>
#include <queue>
using namespace std;
```

[14]:
```
queue<string> bank_queue;
bank_queue.push("Alice");
bank_queue.push("Bob");
bank_queue.push("Charlie");
```

[16]:
```
cout << bank_queue.front();
```

```
Alice
```

[17]:
```
bank_queue
```

[17]:
```
@0x7fa4d67ec0d0
```

[23]:
```
bank_queue.size()
```

[23]:
```
3
```

[21]:
```
bank_queue.empty()
```

[21]:
```
false
```

[24]:
```
// processing the queue
while (!bank_queue.empty())
{
    cout << "Serving: " << bank_queue.front() << endl;
    bank_queue.pop();
}
```

```
Serving: Alice
Serving: Bob
Serving: Charlie
```

1.5 Priority Queue

- A priority queue is a special type of queue where each element has a priority associated with it.
- Elements with higher priority are served before elements with lower priority.
- Common operations: `push()`, `pop()`, `top()`, `empty()`, `size()`
- Example usage:

[ ]:
```
#include <iostream>
#include <queue> // priority_queue is defined in <queue>
#include <string>
#include <pair>
using namespace std;
```

[3]:
```
priority_queue<pair<int, string>> task_queue;
```

[4]:
```
task_queue.push({2, "Low priority task"});
task_queue.push({5, "Medium priority task"});
task_queue.push({10, "High priority task"});
task_queue.push({100, "Critical priority task"});
```

[5]:
```
task_queue.size()
```

[5]: 4

[6]:
```
task_queue.empty()
```

[6]: false

[ ]:
```
cout << task_queue.top().second << endl;
```

```
Critical priority task
```

[8]:
```
// processing the priority queue
while (!task_queue.empty())
{
    cout << "Processing: " << task_queue.top().second << " with priority " <<␣
  ↪task_queue.top().first << endl;
    task_queue.pop();
}
```

```
Processing: Critical priority task with priority 100
Processing: High priority task with priority 10
Processing: Medium priority task with priority 5
Processing: Low priority task with priority 2
```

1.6 Kattis Problems for Demo

- Slide Puzzle - https://open.kattis.com/problems/sliderpuzzle
  - queue, breath first search (BFS) to check if the puzzle is solvable

1.7 Kattis Problems

- Knigs of the Forest - https://open.kattis.com/problems/knigsoftheforest
  - priority_queue
- Select Group - https://open.kattis.com/problems/selectgroup
  - stack and set
- Jane Eyre - https://open.kattis.com/problems/janeeyre
  - simulate using Priority Queue and a sorted list of gifts or two sorted lists of books less than Jane Eyre
- Bank Closing - https://open.kattis.com/problems/bankclosing
  - min priority queue to simulate the process