Admin

- Today’s topics
  - Stack, Queue implementations
  - Start Editor Buffer case study
- Reading
  - Ch 10 & 9

Text editor case study

- Command-line text editing commands
  - F: Move cursor forward one character position
  - B: Move cursor backward one character position
  - J: Jump to start of buffer (before first character)
  - E: Move cursor to end of buffer (after last character)
  - Ixxx: Insert characters xxx at current cursor position
  - D: Delete character after current cursor position

- Buffer requirements
  - Sequence of characters + cursor position
  - Operations to match commands above

- What to learn?
  - Implementation choices, performance implications

Buffer class interface

```cpp
class Buffer {
    public:
        Buffer();
        ~Buffer();
        void moveCursorForward();
        void moveCursorBackward();
        void moveCursorToStart();
        void moveCursorToEnd();
        void insertCharacter(char ch);
        void deleteCharacter();
        void display();
    private:
        // TBD!
};
```

Buffer layered on Vector

- Need character data + cursor
  - Chars in std::vector<char>
  - Represent cursor as integer index
    - Minor detail -- is index before/after cursor?

- Buffer contains: AB|CDE

```
// for Buffer class
private:
    Vector<char> chars;
    int cursor;
```

```
// Buffer contains AB|CDE
A B C D E
  2
```
Evaluate Vector Buffer

Buffer() \quad O(1)
~Buffer() \quad O(1)
moveCursorForward() \quad O(1)
moveCursorBackward() \quad O(1)
moveCursorToStart() \quad O(1)
moveCursorToEnd() \quad O(1)
insertCharacter() \quad O(N)
deleteCharacter() \quad O(N)
Space used \quad \sim 1 \text{ byte per char}

Buffer layered on Stack

\begin{itemize}
  \item Inspiration: add/remove at end of vector is fast
    \begin{itemize}
      \item If chars next to cursor were at end…
        \begin{itemize}
          \item Build on top of stack!
          \item Another layered abstraction!
          \item How is cursor represented?
        \end{itemize}
    \end{itemize}
  \item Buffer contains: AB|CDE
\end{itemize}

// for Buffer class
private:
  Stack<char> before, after;

Buffer as linked list

\begin{itemize}
  \item Inspiration: contiguous memory is constraining
    \begin{itemize}
      \item Connect chars without locality
      \item Linked list to the rescue!
    \end{itemize}
  \item Buffer contains: AB|CDE
\end{itemize}

// for Buffer class
private:
  struct cellT {
    char ch;
    cellT *next;
  };
  cellT *head, *cursor;

Compare implementations

<table>
<thead>
<tr>
<th></th>
<th>Vector</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer()</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>~Buffer()</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>moveCursorForward()</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>moveCursorBackward()</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
<tr>
<td>moveCursorToStart()</td>
<td>O(1)</td>
<td>O(N)</td>
</tr>
<tr>
<td>moveCursorToEnd()</td>
<td>O(1)</td>
<td>O(N)</td>
</tr>
<tr>
<td>insertCharacter()</td>
<td>O(N)</td>
<td>O(1)</td>
</tr>
<tr>
<td>deleteCharacter()</td>
<td>O(N)</td>
<td>O(1)</td>
</tr>
<tr>
<td>Space used</td>
<td>1N</td>
<td>2N</td>
</tr>
</tbody>
</table>