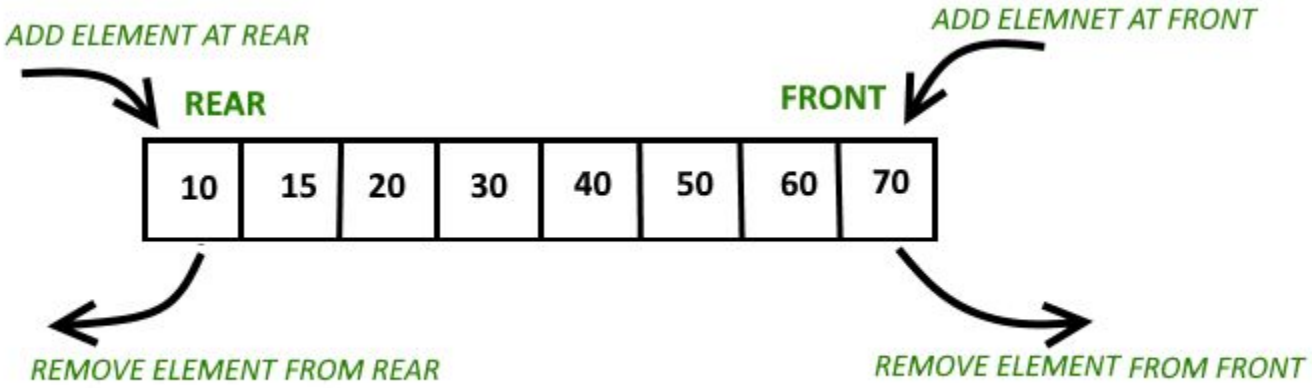
Lab 8

Partnering Up

- Each partner does the **same** provided lab for the week
  - **Not** working together, both partners need to **individually** complete and demo the lab
- Still need to write test cases and report for partner like normal
- Don’t forget to turn in the correct files
  - Test file that you personally wrote
  - Report that you personally wrote
  - Do **not** need to turn in files that your partner wrote
- One more opportunity to turn in the correct files
  - Resubmit any files (if need be) **by the end of lab today** to get credit for your report/test file

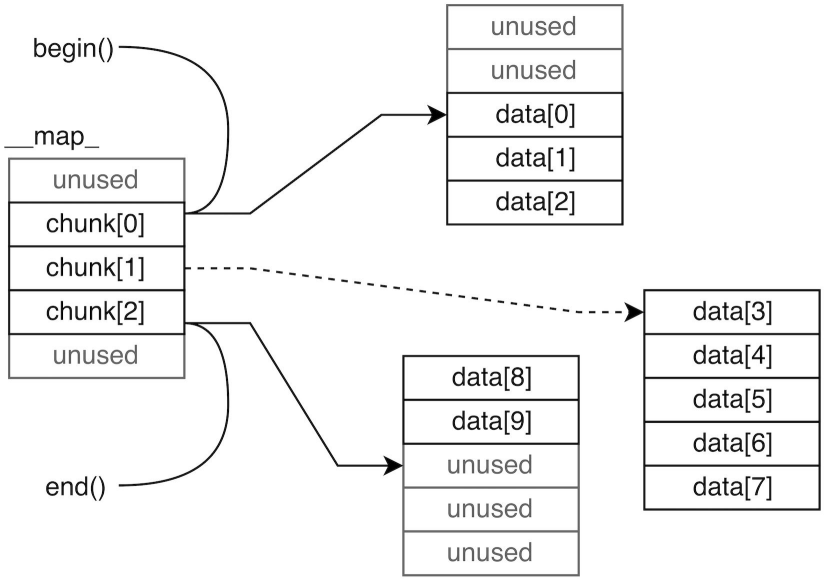
Deque



- Double ended queue
- Insert and remove from either side of the data structure

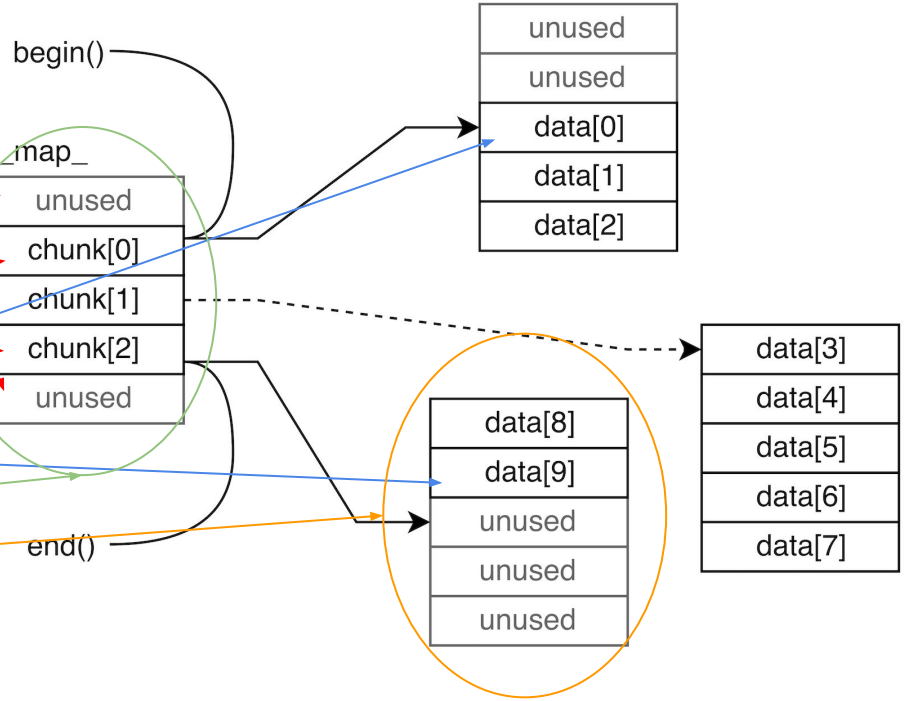
STL Deque

- Pointer to an array of Item pointers
- Can think of the data blocks as being “linked” to create one deque



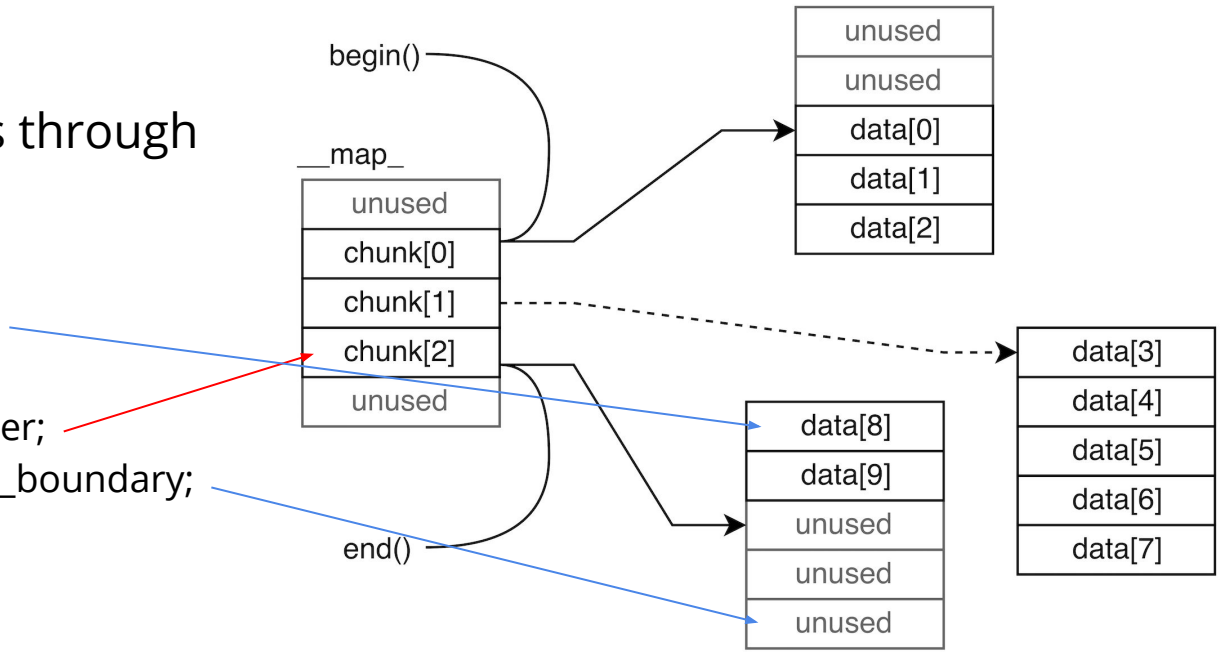
Deque

- Private variables
  - value_type** blocker_pointers;
  - value_type** blocker_pointer_end;
  - value_type** first_bp;
  - value_type** last_bp;
  - value_type* front_ptr;
  - value_type* back_ptr;
  - size_type bp_array_size;
  - size_type block_size;



Deque Iterator

- Pointer that iterates through your deque
- Private variables
  - value_type* cursor;
  - value_type** current_block_pointer;
  - value_type* current_boundary;



Help/Notes

- Reserve() adds slots to both side of the deque
  - Important for push()
- Read the comments, they are there to help guide you through the lab
- **Add this line to copy constructor of deque**
  - bp_array_size = 0;
- Write deque first and then deque_iterator
  - push() and pop() first

Provided Files

- Deque
  - deque.h
    - Will implement your deque in this class
  - deque_iterator.h
    - Will implement your deque_iterator in this class
  - deque_test.cpp
    - Test file
- No need to create any files

Compile/Demo

- Only one cpp file
  - g++ deque_test.cpp
  - ./a.out
- No official test file for demo
  - Program needs to run through 10 tests

Don’t forget

- Demo code to me
  - Either today or next week
  - **Must compile and run on linux servers**
- Submit code to camino by the end of next lab
- Comment code
  - Loops and conditionals
- File with description of lab is on Camino