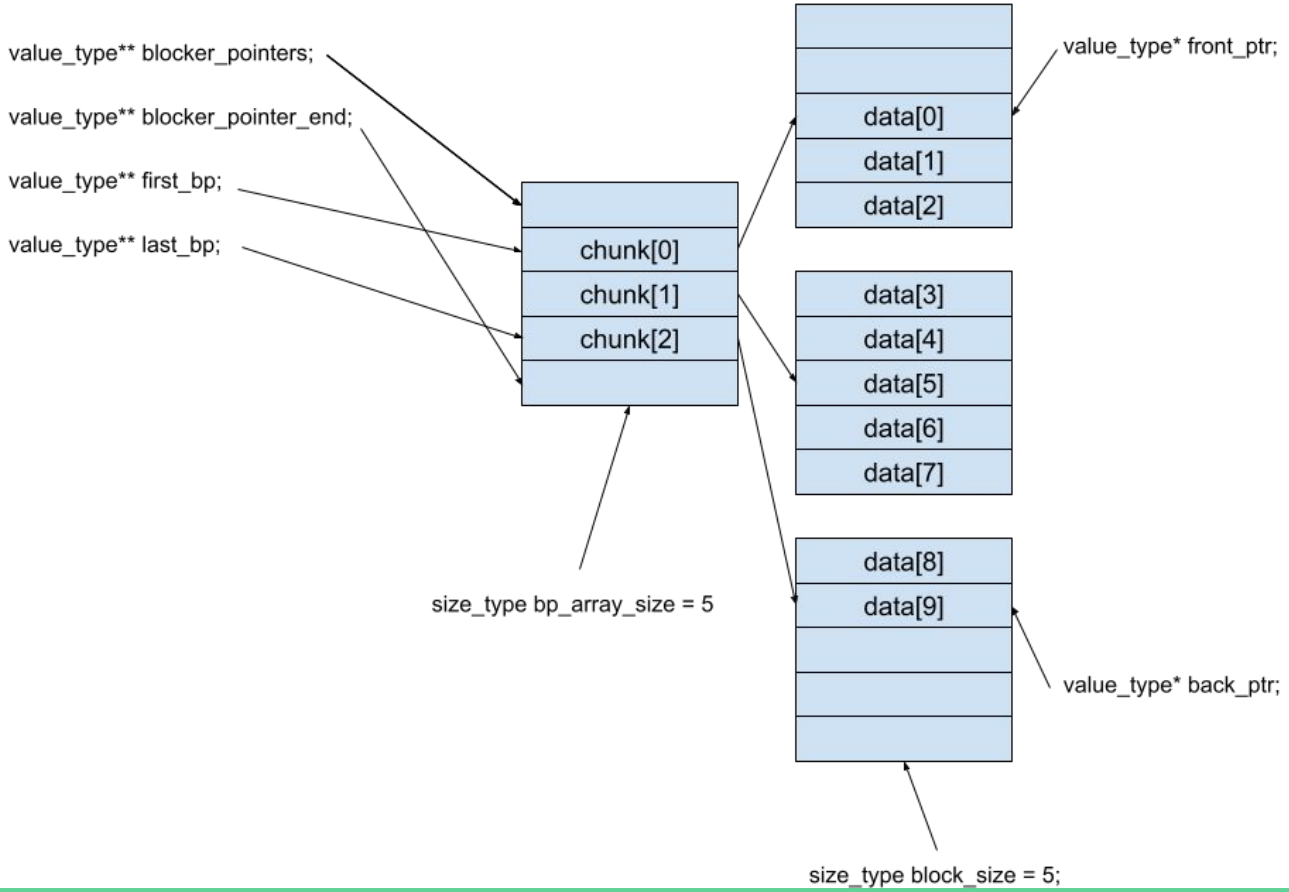
Lab 8

Deque

- Stack + Queue
  - Add/remove front; add/remove end. No insertion/deletion in the middle.
  - For this lab you have a slightly different variation

Deque.h

- Private Variables
  - value_type** blocker_pointers; // Pointer to front of the entire chunk
  - value_type** blocker_pointer_end; // Pointer to end of the entire chunk
  - value_type** first_bp; // Pointer to front of the existing pointer
  - value_type** last_bp; // Pointer to end of the existing pointer
  - value_type* front_ptr; // Pointer to front of the existing data
  - value_type* back_ptr; // Pointer to end of the existing data
  - size_type bp_array_size; // Capacity for the chunks
  - size_type block_size; // Capacity of each block


value_type** blocker_pointers;
value_type** blocker_pointer_end;
value_type** first_bp;
value_type** last_bp;
chunk[0]
chunk[1]
chunk[2]
size_type bp_array_size = 5
data[0]
data[1]
data[2]
value_type* front_ptr;
data[3]
data[4]
data[5]
data[6]
data[7]
data[8]
data[9]
value_type* back_ptr;
size_type block_size = 5;

Deque_iterator.h

- Iterates Through the Deque
- Private Variables
  - value_type* cursor; // Pointer to the data
  - value_type** current_block_pointer; // Pointer to the chunk
  - value_type* current_boundary; // Pointer to the very end of the last block

Hints

- Reserve() adds slots for the deque
  - push()
- Read the comments
- Implement deque.h first, then deque_iterator.h
  - push() and pop()

Provided Files

- deque.h
- deque_iterator.h
- Write your implementations in the .h file
  - Style recommendation (not required for lab)
    - When you have to write implementation details inside .h files (i.e. templates), add “_impl” suffix.

Don’t Forget

- Demo code to me
  - Today or next week
  - Must compile and run on linux servers
- Submit your code before deadline
- Comment code
- File with description of lab is on Camino
  - Submission guidelines