

Double ended Queues

Dequeues are declared as an "abstract" class. They are currently implemented in two ways.

XPDeque

implement dynamically-sized Deques via XPlexes.

DLDeque

implement dynamically-size Deques via linked lists.

All possess the same capabilities. They differ only in constructors. XPDeque constructors optionally take a chunk size argument. DLDeque constructors take no argument.

Double-ended queues support both stack-like and queue-like capabilities:

Assume the declaration of a base element **x**.

Deque d; or **Deque d(int initial_capacity)** declares a deque.

d.empty() returns true if deque d is empty.

| | |
|-----------------|---|
| d.full() | returns true if deque d is full. Always returns false in current implementations. |
|-----------------|---|

d.length() returns the current number of elements in the deque.

d.eng(x) inserts x at the rear of deque d.

| | |
|------------------|------------------------------------|
| d.push(x) | inserts x at the front of deque d. |
|------------------|------------------------------------|

| | |
|--------------------|---|
| x = d.deg() | dequeues and returns the front of deque |
|--------------------|---|

d.front() returns a reference to the front of deque.

d.rear()

returns a reference to the rear of the deque.

d.del_front()

deletes, but does not return the front of deque

d.del_rear()

deletes, but does not return the rear of the deque.

d.clear()

removes all elements from the deque.