Alex Lehmann

Com Sci 202

**Stack Array implementation analysis**

Stack();

O(1) : no loops

Stack(const Stack &obj);

O(n) : loops through list once to set the new array to all the right values

~Stack();

O(n) : loops through array once in order to delete all the elements

void push(const E& e);

O(1) : While it does have a loop it only needs to run that loop every n times so the amount of data used rounds out

const E& pop();

O(1) : no loops

const E& top() const;

O(1) : no loops

int size() const;

O(1) : no loops

bool empty() const;

O(1) : no loops

Overall this is a pretty efficient program as only two methods are linear and they are probably methods that are going to be used less.

**Stack Linked List implementation analysis**

Stack();

O(1) : no loops

Stack(const Stack &obj);

O(n) : loops through list once to set the new list to all the right values

~Stack();

O(n) : loops through array once in order to delete all the elements

void push(const E& e);

O(1) : no loops

const E& pop();

O(1) : no loops

const E& top() const;

O(1) : no loops

int size() const;

O(1) : no loops

bool empty() const;

O(1) : no loops

Like the array implementation this is a pretty efficient program as only two methods are linear and they are probably methods that are going to be used less.

**Queue Array implementation analysis**

Queue();

O(1) : no loops

Queue(const Queue &obj);

O(n) : loops through list once to set the new list to all the right values

~Queue();

O(n) : loops through array once in order to delete all the elements

void enqueue(const E& e);

O(1) : While it does have a loop it only needs to run that loop every n times so the amount of data used rounds out

const E& dequeue();

O(1) : no loops

const E& front() const;

O(1) : no loops

int size() const;

O(1) : no loops

bool empty() const;

O(1) : no loops

This implementation has the same efficiency as the stack implementations. Only the copy constructor and destructors have a linear efficiency.

**Queue Linked List implementation analysis**

Queue();

O(1) : no loops

Queue(const Queue &obj);

O(n) : loops through list once to set the new list to all the right values

~Queue();

O(n) : loops through array once in order to delete all the elements

void enqueue(const E& e);

O(1) : no loops

const E& dequeue();

O(1) : no loops

const E& front() const;

O(1) : no loops

int size() const;

O(1) : no loops

bool empty() const;

O(1) : no loops

Once again this one has the same efficiency as all the rest making it pretty effective.