CSC215

Math and Computer Science

SOUTH DAKOTA
SCHOOL OF MINES
& TECHNOLOGY

Sorted List

- Move to integers (makes it easier to grasp concept)
- Put the numbers in order into list (ascending)

Data Abstraction

Create

Destroy

Insert an item

Remove an item

Find an item

Retrieve an item

See if list is empty

Count the number of items

Print the list

Functional Abstraction

- Create – create an empty list

  In: nothing out: nothing

- Destroy

  In: nothing out: nothing

Functional Abstraction

Insert – Insert an item

in: item out: T/F flag for success

Remove – remove an item

in: item out: T/F flag for success

Functional Abstraction

- Find an item in the list

  in: item out: T/F flag and position

- See if list is empty

  in: nothing out: T/F flag

- Retrieve number of items in list

  in: nothing out: count of items in list

- Print the list

  in: ostream out: nothing

The Sorted List Definition (.h File)

```
class sortedList
{   public:
        sortedList();
        ~sortedList();

        bool insert( int item );
        bool remove( int item );

        int find( int item );
        bool isEmpty();
        int size();
        void print( ostream &out );
};      * no private data yet.
```

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Array Based List drawbacks

- Add, move all items necessary down 1 element in list

Before:	Grocery	Eval	Grade	Laundry	
After:	Grocery	Eval	Room for newitem	Grade	Laundry

- Remove, move all items necessary up 1 element in the list

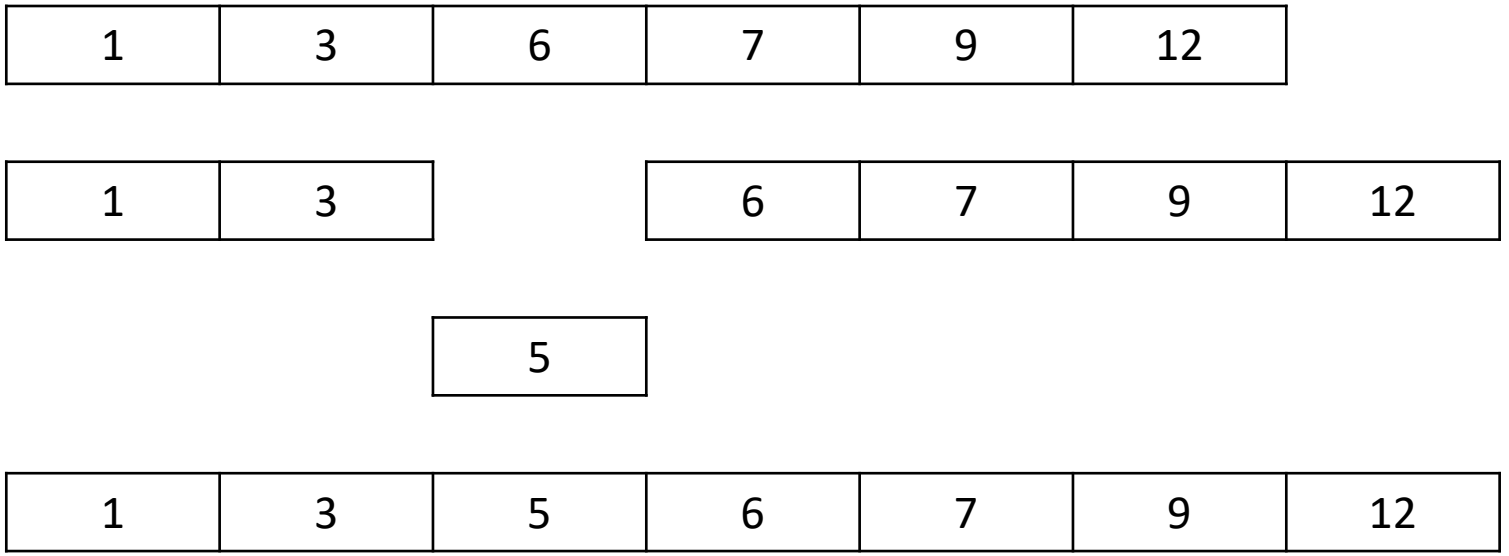
Before:	Grocery	Eval	Grade	Laundry	
After:	Grocery	Eval	Laundry		

- The list doesn't grow and shrink as needed. Yes, dynamic allocation would get around this problem

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example - Insert

- List of integers and I want to put 5 into the list

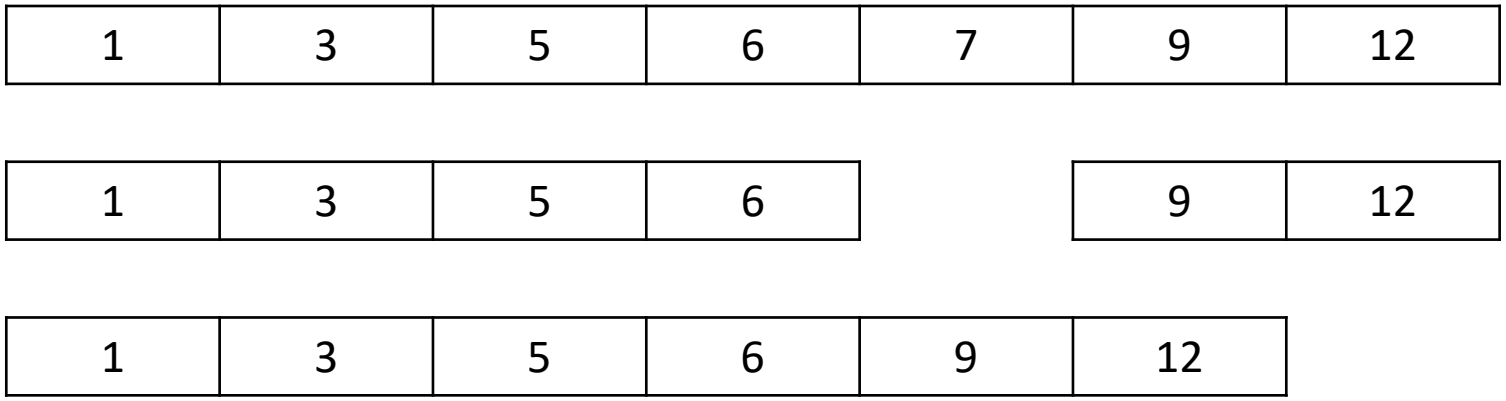


- No more moving data around

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY

Example - Remove

- List of integers and I want to remove 7 into the list



- No more moving data around

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY