Data Structures Midterm Answer

By Po-Chuan & Pei-Hsuan

2015/12/28

Problem 2

(a)Using these operations, write statements (in pseudocode) to compute the sum of two polynomials

```
polynomial sum
high = max( a.degree(), b.degree() )
for ( i from 0 to high )
   sum.changeCoefficient(
      a.coefficient( i ) + b.coefficient( i ), i )
```

Problem 2

(b) Can one find out the highest possible degree of a polynomial supported by a particular implementation of the ADT? How?

Yes.

Suppose we have a polynomial P, one can call `changeCoefficient( P.power() + 1, 1 )` until the function fails (returns false).

When the call fails, `P.power()` returns the highest supported degree of the polynomial.

Problem 4

When the number of items is no greater than half of the current bag capacity, cut down the bag capacity to its half. Rewrite `remove()` to add this feature.

Problem 4

```
template<typename ItemType>
bool ArrayBag::remove( const ItemType& anEntry )
{
   int locatedIndex = getIndexOf( anEntry );
   bool canRemoveItem = ( locatedIndex > -1 );

   if ( canRemoveItem )
      items[ locatedIndex ] = items[ --itemCount ];
```

Problem 4

```
    if ( itemCount <= maxItem / 2 )
    {
        ItemType* N = new ItemType[ maxItem /= 2 ];
        copy( items, item + itemCount, N );
        delete[] items;
        items = N;
    }

    return canRemoveItem;
}
```

Problem 5

Rewrite this function to add an item at the ending position of our bag.

```
template<class ItemType>
bool LinkedBag<ItemType>::add_end(const ItemType& newEntry)
{
    Node<ItemType>* newNodePtr = new Node<ItemType>();
    Node<ItemType>* curPtr = headPtr;
    newNodePtr->setItem( newEntry );
    newNodePtr->setNext( nullptr );
```

Problem 5

Rewrite this function to add an item at the ending position of our bag.

```
    while ( curPtr->getNext() != nullptr )
        curPtr = curPtr->getNext();
    curptr->setNext( newNodePtr );
    itemCount++;

    return true;
}
```

Problem 6

(a) What will be output of the following program?

```
f<double, double>(1.23, 10.3);          11.53
f<double, int>(1.23, 10.3);             11.23
f<int, double>(1.23, 10.3);             11.3
f<char, char>(1.23, 10.3);              11
```

Problem 6

(b) Suppose that we have two classes A and B, and want to use one bag to store items of both classes. Propose a way to modify LinkedBag or A and B to complete the task.

Problem 6

```
template<typename A, typename B>
class LinkedBag
{
      private:
            A* listA;
            B* listB;

      public:
            add( const A ) { /* Codes for add */ }
            add( const B ) { /* Codes for add */ }

};
```

Problem 7

Show the status of the stack and the current postfix expression after each character of the infix expression is processed.

Character	Stack contents	Postfix expr.
(	(	
a	(	a
/	(/	a
(	(/(	a
b	(/(	ab
/	(/(/	ab
c	(/(/	abc
)	(/	abc/

Problem 7

Show the status of the stack and the current postfix expression after each character of the infix expression is processed.

Character	Stack content	Postfix expr.
)		abc//
+	+	abc//
d	+	abc//d
-	-	abc//d+
e	-	abc//d+e
*	-*	abc//d+e
f	-*	abc//d+ef
		abc//d+ef*-

Problem 10

Design an algorithm (in pseudocode) to sort a stack of interger in increasing order with the smallest element on the top.

```
S: initial stack
A: temporary stack A
B: temporary stack B

move S into A
while ( A is not empty )
{
      move A into B, record the max element of A, called M
      move B to S (if this element = M) or A (otherwise)
}
```

Problem 10

Design an algorithm (in pseudocode) to sort a stack of interger in increasing order with the smallest element on the top.

```
S: initial stack
T: temporary stack

move S into T, and record the max element, called M
while ( T is not empty )
{
        push M into S
        move T to S (if this element != M)
        move S to T, but stop at M, and record the max
                element, N
        M = N
}
```

The end