习题

4.18

void reverse(Queue& Q1, Stack& ST1)

{

int length = Q1.length();

while (Q1.length() != 0)

ST1.push(Q1.dequene);

while (Q1.length() != length)

Q1.enquene(ST1.pop());

}

项目设计

4.1

尾部删除:

E dequeue\_tail() { // Take element out

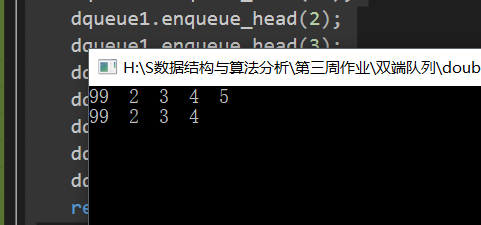
assert(length() != 0, "Queue is empty");

E it = listArray[rear];

rear = (rear - 1) % maxSize; // Circular increment

return it;

}



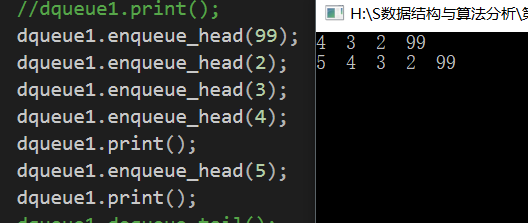
头部添加:

void enqueue\_head(const E& it) { // Put "it" in queue

listArray[front - 1] = it;

front = front - 1;

}



习题:

5.5

<a>

template <typename E>

void inorder(BinNode<E>\* root)

{

if (root == NULL)return;

inorder(root->left());

visit(root);

inorder(root->right());

}

<b>

template <typename E>

void postorder(BinNode<E>\* root)

{

if (root == NULL)return;

postorder(root->left());

postorder(root->right());

visit(root);

}

5.6

template <typename E>

bool search(BinNode<E>\* root, Key K)

{

if (root == NULL)

return false;

if (root->value() == K)

return true;

search(root->left(), K);

search(root->right(), K);

}

附加题:

非递归后序遍历

输入a(b(,d),c(e(g,),f(h,i)))

void postorder(BinTree \*root)

{

stack<BinTree\*> s;

BinTree \*cur;

BinTree \*pre = NULL;

s.push(root);

while (!s.empty())

{

cur = s.top();

if ((cur->lchild == NULL&&cur->rchild == NULL) ||

(pre != NULL && (pre == cur->lchild || pre == cur->rchild)))

{

cout << cur->data << " ";

s.pop();

pre = cur;

}

else

{

if (cur->rchild != NULL)

s.push(cur->rchild);

if (cur->lchild != NULL)

s.push(cur->lchild);

}

}

}

