

第3章

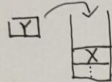


中国海洋大学
OCEAN UNIVERSITY OF CHINA

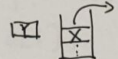
3.4 (1) 逆置栈中元素 (2) 删除栈中元素 e (如果有)

3.5 准则: 序列的任意前 n 个数中, S 数大于等于 X 数

证明: 设第 t 位不同, 此时栈顶为 X , 待输入为 Y
序列 1 的 t 位为 S , 序列 2 的 t 位为 X



序列 1 t 位操作



序列 2 t 位操作



可以看到, 由序列 1 得到的输出序列为 $\dots YX \dots$
由序列 2 得到的输出序列为 $\dots XY \dots$
所以得证。

3.6 输入序列为递增序列。因为 $i < j$ 且 $i < k$, 而 $p_i > p_j$ 且 $p_i > p_k$, 即先输出 p_i , 因此 p_i 出栈输出时, p_j 和 p_k 必定在栈中。又因为输入序列为递增的, 所以肯定是小的先入栈, 大的再入栈, 大的先出栈, 小的再出栈, 故不可能出现 $p_j < p_k$

示内容

```
3.9 void digui (int n)
{ if (n > 1)
  { printf ("%d", n--);
    digui (n);
  }
  return;
}
```



中国海洋大学
OCEAN UNIVERSITY OF CHINA

3.17 bool JudgeStr(char a[])

```
{ stack st;
  InitStack(st);
  int i=0;
  while(a[i]!='&')
  { Push(st, a[i]);
    i++;
  }
  i+=1;
  ElemType temp;
  while(a[i]!='@')
  { Pop(st, x)
    if(x!=a[i])
    { DestroyStack(st);
      return FALSE;
    }
    i++;
  }
  return TRUE;
}
```



```
3. 19 bool Judge(char a[])
{
    stack st;
    InitStack(st);
    int i = 0;
    ElemType t;
    while (a[i])
    {
        if (a[i] == '(' || a[i] == '[' || a[i] == '{')
            Push(st, a[i]);
        else if (a[i] == ')' || a[i] == ']' || a[i] == '}')
        {
            GetTop(st, t);
            if (t != a[i])
            {
                DestroyStack(st);
                return FALSE;
            }
            else { Pop(st, t); }
        }
        else { }
        i++;
    }
    return TRUE;
}
```



3.21 void transfc(char a[])

```
{ Stack st;
  InitStack(st);
  ElemType x;
  int i=0, j=0;
  char t[100];
  while(a[i])
  { if (!IsOperator(a[i]))
    { t[j++] = a[i];
    } else { if (!StackEmpty(st))
    { Push(st, a[i]); }
    else
    { GetTop(st, x);
      if (Pior(a[i], x))
      { Push(st, a[i]); }
      else
      { while (!StackEmpty(st) && !Pior(a[i], x))
        { Pop(st, x);
          t[j++] = x;
          GetTop(st, x);
        }
      }
    }
  }
  }
  i++;
  }
  a = t;
}
```

接下一页



中国海洋大学
OCEAN UNIVERSITY OF CHINA

```
bool IsOperator (char c)
{ if (c == '+' || c == '-' || c == '*' || c == '/' || c == '%')
    return TRUE;
  else
    return FALSE;
}
```

```
bool Prior (char c1, char c2)
{ int t1, t2;
  if (c1 == '+' || c1 == '-')
    t1 = 1;
  else
    t1 = 0;
  if (c2 == '+' || c2 == '-')
    t2 = 1;
  else
    t2 = 0;
  return t1 > t2; // 相同优先级, 由于从左向右运算, 故此处取大于号
}
```



中国海洋大学

OCEAN UNIVERSITY OF CHINA

3.3 3.3.1 bool Judgement (char a[])

```
{ Stack st;
  InitStack (st);
  ElemType x;
  int i = 0;
  while (i < a.Length / 2)
  { Push (st, a[i]);
    i++;
  }
  if (a.Length % 2)
    i += 1;
  while (a[i] != '@' && !StackEmpty(st))
  { Pop (st, x);
    if (!x == a[i])
    { DestroyStack (st);
      return FALSE;
    }
  }
  return TRUE;
}
```



3.32

```
int Fibonacci (int k, int n)
{ if (k < 1) exit (OVERFLOW);
  Queue q;
  InitQueue (q, k);
  ElemType x, e;
  int i = 0;
  while (i <= n)
  { if (i < k-1)
    { if (!EnQueue (q, 0)) exit (OVERFLOW); }
    if (i == k-1)
    { if (!EnQueue (q, 1)) exit (OVERFLOW); }
    if (i >= k)
    { x = sum(q);
      DeQueue (q, e);
      EnQueue (q, x);
    }
    i++;
  }
  return q.base[(q.rear + q.MaxSize - 1) % q.MaxSize];
}
```