四川師範大學
SICHUAN NORMAL UNIVERSITY

第3章 栈和队列

DATA STRUCTURE

计算机科学学院 刘 芳

四川師範大學
SICHUAN NORMAL UNIVERSITY

第3章 栈和队列

3.1 栈
3.2 栈的应用举例
3.3 栈与递归
3.4 队列

计算机科学学院　刘 芳

3.2 栈的应用举例

- 3.2.1 数制转换
- 3.2.2 括号匹配的检查
- 3.2.3 表达式求值

刘　芳 LiuFang

四川師範大學
SICHUAN NORMAL UNIVERSITY

3.2.1 数制转换

刘　芳 LiuFang

四川師範大學
SICHUAN NORMAL UNIVERSITY

问题提出与分析

- 将十进制整数N转换为r进制。

$$(N)_{10}=(d_{n-1}\cdots\cdots d_1 d_0)_r$$

$$=d_{n-1}\times r^{n-1}+\cdots\cdots+d_1\times r^1+d_0\times r^0$$

- 分析：
  - 除基倒序取余法
- 例如
  - $(3425)_{10}=(\ 6541\ )_8$

8 | 3425		余数	
8 | 428	.....	1	d_0
8 | 53	.....	4	d_1
8 | 6	.....	5	d_2
0	.....	6	d_3

练一练

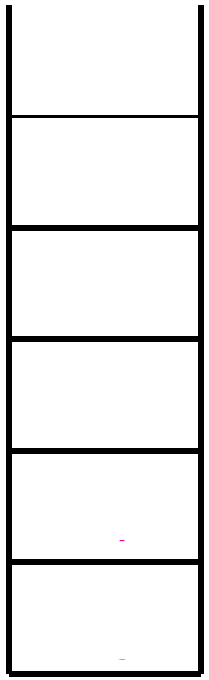
- $(1348\ )_{10}=(\ 2504\ \ \ \ )_8$
- $(100\ \ \ )_{10}=(110\ 0100)_2$
- $(181\ \ \ )_{10}=(\ B5\ \ \ \ \ \ )_{16}$

算法思路

- 借助栈的LIFO的特性

$$(N)_{10}=(d_{n-1}\ldots\ldots d_1 d_0)_r$$

$$=d_{n-1}\times r^{n-1}+\ldots\ldots+d_1\times r^1+d_0\times r^0$$



d_{n-1} …… d_1 d_0

算法描述

```
void conversion(int N,int r) {
        InitStack(S);
        while (N){
            Push(S, N%r);
            N=N/r;
        }
        while(!StackEmpty(S)){
            Pop (S,e);
            if (e<10) printf("%d",e);
            else printf("%c",'A'+e-10);
        }
}
```

1.初始化空栈

2.余数依次压栈

3.依次出栈，并输出。（倒序取余）

本节小结

- 十进制整数转换为r进制
  - 转换规则：除基倒序取余法
  - 数据结构：栈
- 思考：
  - 如何实现任意进制的整数的转换？
  - 如何实现十进制小数转换为r进制呢？

四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听

业精于勤,荒于嬉;行成于思,毁于随.

3.2 栈的应用举例

- 3.2.1 数制转换
- 3.2.2 括号匹配的检查
- 3.2.3 表达式求值

刘　芳 LiuFang

四川師範大學
SICHUAN NORMAL UNIVERSITY

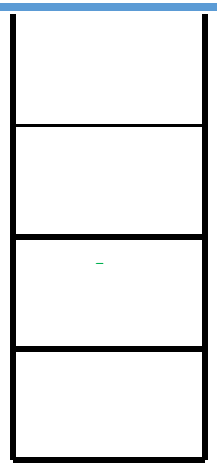
3.2.2 括号匹配的检查

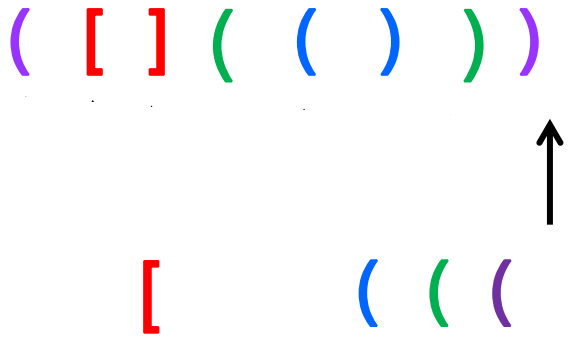
刘　芳 LiuFang

四川師範大學
SICHUAN NORMAL UNIVERSITY

问题提出与分析

- ([ ] (( ))) ✔
- [(( ]) ✘
- [(( ) ✘
- (( ))]) ✘





算法描述

思考：若还有{ } < > 等呢？

```
Status Compare(char str[ ] ){
    InitStack(S); i=0; flag=TRUE;
    while (str[i]!='\0' && flag ){
        switch (str[i]) {
          case '(' :
          case '[' :Push(S,str[i]);break;
          case ')' :if (Pop(S,e)==ERROR  || e!='(' )  flag=FALSE;break;
          case ']' :if (Pop(S,e)==ERROR  || e!='[' )  flag=FALSE;break;
        }
        i++;
    }
  if ( flag &&(str[i]=='\0') && StackEmpty(S))  return TRUE;
  else return FALSE;
}
```

四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听

业精于勤,荒于嬉;行成于思,毁于随.

3.2 栈的应用举例

- 3.2.1 数制转换
- 3.2.2 括号匹配的检查
- 3.2.3 表达式求值

刘　芳 LiuFang

四川師範大學
SICHUAN NORMAL UNIVERSITY

3.2.3 表达式求值

刘　芳 LiuFang

四川師範大學
SICHUAN NORMAL UNIVERSITY

问题提出与分析

- 四则运算表达式的求值
  - 3*(7-2)　　3*(5-2)+7　　(4+2)*3-10/5
- 设：
  - 表达式没有语法错误
  - 运算符：+，-，*，/，( )
  - 只考虑求解次序（运算符的优先级）
- 问题：
  - 算法优先级别？
  - 如何求值？

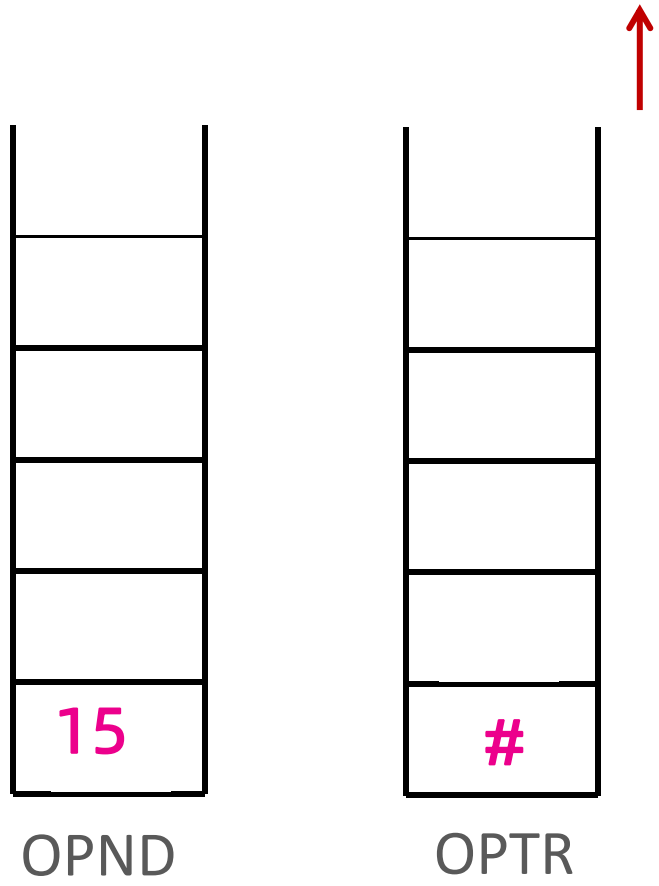
算符间的优先关系

θ_1 \ θ_2	+	-	×	/	(	)	#
+	>	>	<	<	<	>	>
-	>	>	<	<	<	>	>
×	>	>	>	>	<	>	>
/	>	>	>	>	<	>	>
(	<	<	<	<	<	=	error
)	>	>	>	>	error	>	>
#	<	<	<	<	<	error	=

算法思想

- 设置两个工作栈
  - 操作数栈 OPND
  - 操作符栈 OPTR

例如：3 * (7 - 2) #



算法思想

- 自左至右扫描表达式,c为当前字符:
  - c是操作数， Push(OPND,c);
  - c是操作符， Precede(GetTop(OPTR), c)
    1. < : Push(OPTR,c);继续处理下一个字符。
    2. > : Pop(OPTR,θ); Pop(OPND,b);Pop(OPND,a);
       Push(OPND,aθb)
       继续处理当前字符c。
    3. 若相等， Pop(OPTR,x),脱括号， 继续处理下一个字符。
- 直到遇到结束符。

算法描述

```
OperandType Evaluateexpress(){
  InitStack(OPTR); Push(OPTR,'#');   InitStack(OPND); c=getchar( );
  while (c!='#' || GetTop(OPTR)!='#'){
      if （c为操作数）{ Push(OPND,c);c=getchar();}
       else
          switch Precede(GetTop(OPTR),c){
            case '<':Push(OPTR,c);  c=getchar( );break;
            case '=':Pop(OPTR,c);    c=getchar( ); break;
            case '>':Pop(OPTR,theta);Pop(OPND,b);Pop(OPND,a);
                     Push(OPND,Operate(a,theta,b);break;
          }//switch
  }
  return  (GetTop(OPND));
}
```

拓展

- 表达式的三种形式
  - 前缀表达式　*3 - 7 2　- * + 4 2 3 / 10 5
  - 中缀表达式　3*(7-2)　(4+2)*3-10/5
  - 后缀表达式　3 7 2 - *　4 2 + 3 * 10 5 / -
- 思考：
  - 后缀表达式的求值？
  - 中缀表达式→后缀表达式？

四川師範大學
SICHUAN NORMAL UNIVERSITY

感谢聆听

业精于勤,荒于嬉;行成于思,毁于随.