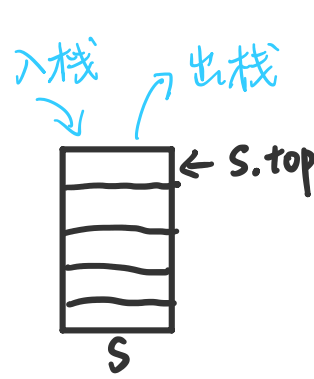


栈

逻辑结构知识点

栈	• 只允许在一端进行插入或删除操作的线性表
栈顶	• 线性表允许进行插入删除的那一段
栈底	• 固定的，不允许进行插入删除的那一段
操作特性	• 先进后出
数学性质	• n个不同元素进栈，出栈元素的不同排列的个数为 $\frac{1}{n+1}C_n^n$ (卡特兰数)



存储结构知识点

1、栈的顺序存储结构															
顺序栈的定义	• 采用顺序存储的栈 • 利用一组地址连续的存储单元存放自栈底到栈顶的数据元素														
顺序栈的实现	<pre>#define MaxSize 50 typedef struct { Elemtype data[MaxSize]; //存放栈中元素 int top; //栈顶指针 } SqStack;</pre> <table><tr><td>栈顶指针</td><td>S.top, 初始时设置S.top=-1; 栈顶元素: S.data[S.top]</td><td></td><td></td></tr><tr><td>进栈操作</td><td>栈不满时, 栈顶指针先加1, 再送值到栈顶元素</td><td>出栈操作</td><td>栈非空时, 先找栈顶元素值, 再将栈顶指针减1</td></tr><tr><td>栈空条件</td><td>S.top== -1</td><td>栈满条件</td><td>S.top==MaxSize-1; 栈长: S.top+1</td></tr></table>			栈顶指针	S.top, 初始时设置S.top=-1; 栈顶元素: S.data[S.top]			进栈操作	栈不满时, 栈顶指针先加1, 再送值到栈顶元素	出栈操作	栈非空时, 先找栈顶元素值, 再将栈顶指针减1	栈空条件	S.top== -1	栈满条件	S.top==MaxSize-1; 栈长: S.top+1
栈顶指针	S.top, 初始时设置S.top=-1; 栈顶元素: S.data[S.top]														
进栈操作	栈不满时, 栈顶指针先加1, 再送值到栈顶元素	出栈操作	栈非空时, 先找栈顶元素值, 再将栈顶指针减1												
栈空条件	S.top== -1	栈满条件	S.top==MaxSize-1; 栈长: S.top+1												
顺序栈基本运算代码	<table><tr><td>初始化</td><td><pre>void InitStack(SqStack &S) { S.top = -1; }</pre></td><td>判栈空</td><td><pre>bool StackEmpty(SqStack S) { if (S.top == -1)//栈空 return true; else return false; }</pre></td></tr><tr><td>进栈</td><td><pre>bool Push(SqStack &S, ElemType x) { if (S.top == MaxSize - 1) //栈满 return false; S.data[++S.top] = x; return ture; }</pre></td><td>出栈</td><td><pre>bool Pop(SqStack &S, ElemType x) { if (S.top == -1) //栈空 return false; S.data[S.top--] = x; return ture; }</pre></td></tr><tr><td>读栈顶元素</td><td><pre>bool GetTop(SqStack S, ElemType x) { if (S.top == -1) //栈空 return false; x = S.data[S.top]; return ture; }</pre></td><td></td><td></td></tr></table>			初始化	<pre>void InitStack(SqStack &S) { S.top = -1; }</pre>	判栈空	<pre>bool StackEmpty(SqStack S) { if (S.top == -1)//栈空 return true; else return false; }</pre>	进栈	<pre>bool Push(SqStack &S, ElemType x) { if (S.top == MaxSize - 1) //栈满 return false; S.data[++S.top] = x; return ture; }</pre>	出栈	<pre>bool Pop(SqStack &S, ElemType x) { if (S.top == -1) //栈空 return false; S.data[S.top--] = x; return ture; }</pre>	读栈顶元素	<pre>bool GetTop(SqStack S, ElemType x) { if (S.top == -1) //栈空 return false; x = S.data[S.top]; return ture; }</pre>		
初始化	<pre>void InitStack(SqStack &S) { S.top = -1; }</pre>	判栈空	<pre>bool StackEmpty(SqStack S) { if (S.top == -1)//栈空 return true; else return false; }</pre>												
进栈	<pre>bool Push(SqStack &S, ElemType x) { if (S.top == MaxSize - 1) //栈满 return false; S.data[++S.top] = x; return ture; }</pre>	出栈	<pre>bool Pop(SqStack &S, ElemType x) { if (S.top == -1) //栈空 return false; S.data[S.top--] = x; return ture; }</pre>												
读栈顶元素	<pre>bool GetTop(SqStack S, ElemType x) { if (S.top == -1) //栈空 return false; x = S.data[S.top]; return ture; }</pre>														
共享栈	<table><tr><td>定义</td><td colspan="3">• 利用栈底位置相对不变的特性, 可让两个顺序栈共享一个一维数组空间 • 将两个栈的栈底分别设置在共享空间的两端, 两个栈顶向共享空间的中间延伸</td></tr><tr><td>特点</td><td colspan="3"><p>① top0=-1时, 0号栈空; top1=MaxSize-1时, 1号栈空 ② top1-top0=1时, 栈满 ③ 0号栈进栈, top0加1, 再放值 ④ 1号栈进栈, top1减1, 再放值</p></td></tr><tr><td>目的</td><td colspan="3">• 更有效地利用存储空间 • 两个栈的空间相互调节, 只有在整个存储空间都被占满时才发生上溢</td></tr></table>			定义	• 利用栈底位置相对不变的特性, 可让两个顺序栈共享一个一维数组空间 • 将两个栈的栈底分别设置在共享空间的两端, 两个栈顶向共享空间的中间延伸			特点	<p>① top0=-1时, 0号栈空; top1=MaxSize-1时, 1号栈空 ② top1-top0=1时, 栈满 ③ 0号栈进栈, top0加1, 再放值 ④ 1号栈进栈, top1减1, 再放值</p>			目的	• 更有效地利用存储空间 • 两个栈的空间相互调节, 只有在整个存储空间都被占满时才发生上溢		
定义	• 利用栈底位置相对不变的特性, 可让两个顺序栈共享一个一维数组空间 • 将两个栈的栈底分别设置在共享空间的两端, 两个栈顶向共享空间的中间延伸														
特点	<p>① top0=-1时, 0号栈空; top1=MaxSize-1时, 1号栈空 ② top1-top0=1时, 栈满 ③ 0号栈进栈, top0加1, 再放值 ④ 1号栈进栈, top1减1, 再放值</p>														
目的	• 更有效地利用存储空间 • 两个栈的空间相互调节, 只有在整个存储空间都被占满时才发生上溢														

2、栈的链式存储结构

概念	• 采用链式存储的栈
优点	• 便于多个栈共享存储空间和提高其效率, 且不存在栈满上溢的情况
特点	• 通常采用单链表实现, 并规定所有操作都是在单链表的表头进行的 • 这里规定链栈没有头结点, Lhead指向栈顶元素
链式栈的实现	<pre>typedef struct LinkNode { Elemtype data; //存放栈中元素 struct LinkNode *next; //栈顶指针 } * LiStack;</pre>

栈的运算/操作

InitStack(&S)	初始化栈	• 初始化一个空栈S
StackEmpty(S)	判断栈是否为空	• 空则返回True
Push(&S,x)	进栈	• 若S未满, 则将x加入使之成为新栈顶
Pop(&S,&x)	出栈	• 若S非空, 则弹出栈顶元素, 并用x返回
GetTop(S,&x)	读栈顶元素	• 若栈非空, 则用x返回栈顶元素
DestroyStack(&S)	销毁栈	• 销毁并释放栈S占用的存储空间

栈的应用

括号匹配	<ul style="list-style-type: none">• 初始设置一个空栈, 顺序读入符号• 若是右括号, 置于栈顶• 若是左括号, 压入栈中• 算法结束时, 栈为空, 否则括号序列不匹配(具体例子见下面考点3的例题)																																					
表达式求值	<div><div><ul style="list-style-type: none">• 考点1: 后缀表达式和正常表达式的相互转换• 考点2: 涉及到编译原理的求值过程</div><div><div>$A+B*(C-D)-E/F$<p>后缀表达式</p>$A\ B\ C\ D\ -\ * \ E\ F\ / \ -$<p>编译原理求值过程</p><table><tr><th>扫描符</th><th>栈内元素</th><th>操作内容</th></tr><tr><td>1</td><td>A</td><td>push</td></tr><tr><td>2</td><td>B</td><td>push</td></tr><tr><td>3</td><td>C</td><td>push</td></tr><tr><td>4</td><td>D</td><td>push</td></tr><tr><td>5</td><td>-</td><td>6.D pop, C-D=>E, E push</td></tr><tr><td>6</td><td>*</td><td>E, B push, A*B=>F, F push</td></tr><tr><td>7</td><td>+</td><td>F, E push, A+B=>G, G push</td></tr><tr><td>8</td><td>E</td><td>push</td></tr><tr><td>9</td><td>F</td><td>push</td></tr><tr><td>10</td><td>/</td><td>E-F pop, E/F=>H, H push</td></tr><tr><td>11</td><td>-</td><td>H, G pop, H-G=>I, I push</td></tr></table></div></div></div> <p>考点3: 中缀到后缀表达式转换的过程(见王道P98)</p> <p>12. 【2014 统考真题】假设栈初始为空, 将中缀表达式 $a*b*(c*d+e)f$ 转换为等价的后缀表达式的过程中, 当扫描到 f 时, 栈中的元素依次是()。</p> <p>A. $a, *$ B. $a, *, +$ C. $a, *, +, *$ D. $a, *, +, *, /$</p> <p>解析: 将中缀表达式转换为后缀表达式, 需要遵循以下规则:</p> <ul style="list-style-type: none">遇到左括号, 压入栈中。遇到右括号, 依次弹出栈中元素, 直到遇到左括号为止, 并将弹出的元素依次输出。遇到运算符, 比较该运算符与栈顶运算符的优先级。如果该运算符的优先级高于栈顶运算符, 则将该运算符压入栈中; 否则, 弹出栈顶运算符并输出, 直到栈顶运算符的优先级低于该运算符为止, 然后将该运算符压入栈中。表达式结束后, 依次弹出栈中所有运算符并输出。 <p>对于表达式 $a*b*(c*d+e)f$, 扫描到 f 时, 栈中的元素依次是 $a, *, +, *$。</p>		扫描符	栈内元素	操作内容	1	A	push	2	B	push	3	C	push	4	D	push	5	-	6.D pop, C-D=>E, E push	6	*	E, B push, A*B=>F, F push	7	+	F, E push, A+B=>G, G push	8	E	push	9	F	push	10	/	E-F pop, E/F=>H, H push	11	-	H, G pop, H-G=>I, I push
扫描符	栈内元素	操作内容																																				
1	A	push																																				
2	B	push																																				
3	C	push																																				
4	D	push																																				
5	-	6.D pop, C-D=>E, E push																																				
6	*	E, B push, A*B=>F, F push																																				
7	+	F, E push, A+B=>G, G push																																				
8	E	push																																				
9	F	push																																				
10	/	E-F pop, E/F=>H, H push																																				
11	-	H, G pop, H-G=>I, I push																																				
递归	<ul style="list-style-type: none">• 在递归调用的过程中, 系统为每一层的返回点、局部变量、传入实参等开辟了递归工作栈来进行数据储存• 递归次数过多容易造成栈溢出等• 将递归算法转换为非递归算法, 通常需要借助栈来实现这种转换, 消除递归并不一定需要栈• 效率不高, 原因是递归调用过程中包含很多重复的计算• 代码简单, 容易理解																																					

队列

逻辑结构知识点

队列	操作受限的线性表
队头	允许删除的一端
队尾	允许插入的一端
操作特性	先进先出

存储结构知识点

队列的顺序存储结构											
顺序队列的定义	• 分配一块连续的存储单元存放队列中的元素 • 设两个指针 <ul style="list-style-type: none">◦ 队头指针front指向队头元素◦ 队尾指针rear指向队尾元素的下一个位置										
顺序队列的实现	<pre>#define MaxSize 50 typedef struct { ElemType data[MaxSize]; int front, rear; } SqQueue;</pre> <table><tr><td>初始状态</td><td>Q.front==Q.rear==0</td><td>队满操作</td><td>• Q.rear==MaxSize不能作为队满的条件 • 只有一个元素仍满足该条件 (假退出)</td></tr><tr><td>进队操作</td><td>队不满时, 先送值到队尾元素, 再将队尾指针加1</td><td>出队操作</td><td>队不空时, 先取队头元素值, 再将队头指针加1</td></tr></table> <div><p>图例:</p><p>1. 初始状态: front=0, rear=0. 队列: []</p><p>2. 进队操作: 元素a入队. rear++ (rear=1). 队列: [a]</p><p>3. 出队操作: 元素a出队. front++ (front=1). 队列: []</p><p>4. 进队操作: 元素b入队. rear++ (rear=2). 队列: [b]</p><p>5. 出队操作: 元素b出队. front++ (front=2). 队列: []</p></div>			初始状态	Q.front==Q.rear==0	队满操作	• Q.rear==MaxSize不能作为队满的条件 • 只有一个元素仍满足该条件 (假退出)	进队操作	队不满时, 先送值到队尾元素, 再将队尾指针加1	出队操作	队不空时, 先取队头元素值, 再将队头指针加1
初始状态	Q.front==Q.rear==0	队满操作	• Q.rear==MaxSize不能作为队满的条件 • 只有一个元素仍满足该条件 (假退出)								
进队操作	队不满时, 先送值到队尾元素, 再将队尾指针加1	出队操作	队不空时, 先取队头元素值, 再将队头指针加1								
循环队列的定义											
循环队列的实现 (以下述方法一为例)	• 把存储队列元素的表从逻辑上视为一个环										
<div><p>图例:</p><p>1. 初始状态: front=0, rear=0. 队列: [a]</p><p>2. 进队操作: 元素b入队. rear++ (rear=1). 队列: [a, b]</p><p>3. 出队操作: 元素a出队. front++ (front=1). 队列: [b]</p><p>4. 进队操作: 元素c入队. rear++ (rear=2). 队列: [b, c]</p><p>5. 出队操作: 元素b出队. front++ (front=2). 队列: [c]</p><p>6. 进队操作: 元素d入队. rear++ (rear=3). 队列: [c, d]</p><p>7. 出队操作: 元素c出队. front++ (front=3). 队列: [d]</p><p>8. 进队操作: 元素e入队. rear++ (rear=4). 队列: [d, e]</p><p>9. 出队操作: 元素d出队. front++ (front=4). 队列: [e]</p><p>10. 进队操作: 元素f入队. rear++ (rear=5). 队列: [e, f]</p><p>11. 出队操作: 元素e出队. front++ (front=5). 队列: [f]</p><p>12. 进队操作: 元素g入队. rear++ (rear=6). 队列: [f, g]</p><p>13. 出队操作: 元素f出队. front++ (front=6). 队列: [g]</p><p>14. 进队操作: 元素h入队. rear++ (rear=7). 队列: [g, h]</p><p>15. 出队操作: 元素g出队. front++ (front=7). 队列: [h]</p><p>16. 进队操作: 元素i入队. rear++ (rear=8). 队列: [h, i]</p><p>17. 出队操作: 元素h出队. front++ (front=8). 队列: [i]</p><p>18. 进队操作: 元素j入队. rear++ (rear=9). 队列: [i, j]</p><p>19. 出队操作: 元素i出队. front++ (front=9). 队列: [j]</p><p>20. 进队操作: 元素k入队. rear++ (rear=10). 队列: [j, k]</p><p>21. 出队操作: 元素j出队. front++ (front=10). 队列: [k]</p><p>22. 进队操作: 元素l入队. rear++ (rear=11). 队列: [k, l]</p><p>23. 出队操作: 元素k出队. front++ (front=11). 队列: [l]</p><p>24. 进队操作: 元素m入队. rear++ (rear=12). 队列: [l, m]</p><p>25. 出队操作: 元素l出队. front++ (front=12). 队列: [m]</p><p>26. 进队操作: 元素n入队. rear++ (rear=13). 队列: [m, n]</p><p>27. 出队操作: 元素m出队. front++ (front=13). 队列: [n]</p><p>28. 进队操作: 元素o入队. rear++ (rear=14). 队列: [n, o]</p><p>29. 出队操作: 元素n出队. front++ (front=14). 队列: [o]</p><p>30. 进队操作: 元素p入队. rear++ (rear=15). 队列: [o, p]</p><p>31. 出队操作: 元素o出队. front++ (front=15). 队列: [p]</p><p>32. 进队操作: 元素q入队. rear++ (rear=16). 队列: [p, q]</p><p>33. 出队操作: 元素p出队. front++ (front=16). 队列: [q]</p><p>34. 进队操作: 元素r入队. rear++ (rear=17). 队列: [q, r]</p><p>35. 出队操作: 元素q出队. front++ (front=17). 队列: [r]</p><p>36. 进队操作: 元素s入队. rear++ (rear=18). 队列: [r, s]</p><p>37. 出队操作: 元素r出队. front++ (front=18). 队列: [s]</p><p>38. 进队操作: 元素t入队. rear++ (rear=19). 队列: [s, t]</p><p>39. 出队操作: 元素s出队. front++ (front=19). 队列: [t]</p><p>40. 进队操作: 元素u入队. rear++ (rear=20). 队列: [t, u]</p><p>41. 出队操作: 元素t出队. front++ (front=20). 队列: [u]</p><p>42. 进队操作: 元素v入队. rear++ (rear=21). 队列: [u, v]</p><p>43. 出队操作: 元素u出队. front++ (front=21). 队列: [v]</p><p>44. 进队操作: 元素w入队. rear++ (rear=22). 队列: [v, w]</p><p>45. 出队操作: 元素v出队. front++ (front=22). 队列: [w]</p><p>46. 进队操作: 元素x入队. rear++ (rear=23). 队列: [w, x]</p><p>47. 出队操作: 元素w出队. front++ (front=23). 队列: [x]</p><p>48. 进队操作: 元素y入队. rear++ (rear=24). 队列: [x, y]</p><p>49. 出队操作: 元素x出队. front++ (front=24). 队列: [y]</p><p>50. 进队操作: 元素z入队. rear++ (rear=25). 队列: [y, z]</p><p>51. 出队操作: 元素y出队. front++ (front=25). 队列: [z]</p><p>52. 进队操作: 元素a入队. rear++ (rear=26). 队列: [z, a]</p><p>53. 出队操作: 元素z出队. front++ (front=26). 队列: [a]</p><p>54. 进队操作: 元素b入队. rear++ (rear=27). 队列: [a, b]</p><p>55. 出队操作: 元素a出队. front++ (front=27). 队列: [b]</p><p>56. 进队操作: 元素c入队. rear++ (rear=28). 队列: [b, c]</p><p>57. 出队操作: 元素b出队. front++ (front=28). 队列: [c]</p><p>58. 进队操作: 元素d入队. rear++ (rear=29). 队列: [c, d]</p><p>59. 出队操作: 元素c出队. front++ (front=29). 队列: [d]</p><p>60. 进队操作: 元素e入队. rear++ (rear=30). 队列: [d, e]</p><p>61. 出队操作: 元素d出队. front++ (front=30). 队列: [e]</p><p>62. 进队操作: 元素f入队. rear++ (rear=31). 队列: [e, f]</p><p>63. 出队操作: 元素e出队. front++ (front=31). 队列: [f]</p><p>64. 进队操作: 元素g入队. rear++ (rear=32). 队列: [f, g]</p><p>65. 出队操作: 元素f出队. front++ (front=32). 队列: [g]</p><p>66. 进队操作: 元素h入队. rear++ (rear=33). 队列: [g, h]</p><p>67. 出队操作: 元素g出队. front++ (front=33). 队列: [h]</p><p>68. 进队操作: 元素i入队. rear++ (rear=34). 队列: [h, i]</p><p>69. 出队操作: 元素h出队. front++ (front=34). 队列: [i]</p><p>70. 进队操作: 元素j入队. rear++ (rear=35). 队列: [i, j]</p><p>71. 出队操作: 元素i出队. front++ (front=35). 队列: [j]</p><p>72. 进队操作: 元素k入队. rear++ (rear=36). 队列: [j, k]</p><p>73. 出队操作: 元素j出队. front++ (front=36). 队列: [k]</p><p>74. 进队操作: 元素l入队. rear++ (rear=37). 队列: [k, l]</p><p>75. 出队操作: 元素k出队. front++ (front=37). 队列: [l]</p><p>76. 进队操作: 元素m入队. rear++ (rear=38). 队列: [l, m]</p><p>77. 出队操作: 元素l出队. front++ (front=38). 队列: [m]</p><p>78. 进队操作: 元素n入队. rear++ (rear=39). 队列: [m, n]</p><p>79. 出队操作: 元素m出队. front++ (front=39). 队列: [n]</p><p>80. 进队操作: 元素o入队. rear++ (rear=40). 队列: [n, o]</p><p>81. 出队操作: 元素n出队. front++ (front=40). 队列: [o]</p><p>82. 进队操作: 元素p入队. rear++ (rear=41). 队列: [o, p]</p><p>83. 出队操作: 元素o出队. front++ (front=41). 队列: [p]</p><p>84. 进队操作: 元素q入队. rear++ (rear=42). 队列: [p, q]</p><p>85. 出队操作: 元素p出队. front++ (front=42). 队列: [q]</p><p>86. 进队操作: 元素r入队. rear++ (rear=43). 队列: [q, r]</p><p>87. 出队操作: 元素q出队. front++ (front=43). 队列: [r]</p><p>88. 进队操作: 元素s入队. rear++ (rear=44). 队列: [r, s]</p><p>89. 出队操作: 元素r出队. front++ (front=44). 队列: [s]</p><p>90. 进队操作: 元素t入队. rear++ (rear=45). 队列: [s, t]</p><p>91. 出队操作: 元素s出队. front++ (front=45). 队列: [t]</p><p>92. 进队操作: 元素u入队. rear++ (rear=46). 队列: [t, u]</p><p>93. 出队操作: 元素t出队. front++ (front=46). 队列: [u]</p><p>94. 进队操作: 元素v入队. rear++ (rear=47). 队列: [u, v]</p><p>95. 出队操作: 元素u出队. front++ (front=47). 队列: [v]</p><p>96. 进队操作: 元素w入队. rear++ (rear=48). 队列: [v, w]</p><p>97. 出队操作: 元素v出队. front++ (front=48). 队列: [w]</p><p>98. 进队操作: 元素x入队. rear++ (rear=49). 队列: [w, x]</p><p>99. 出队操作: 元素w出队. front++ (front=49). 队列: [x]</p><p>100. 进队操作: 元素y入队. rear++ (rear=50). 队列: [x, y]</p><p>101. 出队操作: 元素x出队. front++ (front=50). 队列: [y]</p><p>102. 进队操作: 元素z入队. rear++ (rear=51). 队列: [y, z]</p><p>103. 出队操作: 元素y出队. front++ (front=51). 队列: [z]</p><p>104. 进队操作: 元素a入队. rear++ (rear=52). 队列: [z, a]</p><p>105. 出队操作: 元素z出队. front++ (front=52). 队列: [a]</p><p>106. 进队操作: 元素b入队. rear++ (rear=53). 队列: [a, b]</p><p>107. 出队操作: 元素a出队. front++ (front=53). 队列: [b]</p><p>108. 进队操作: 元素c入队. rear++ (rear=54). 队列: [b, c]</p><p>109. 出队操作: 元素b出队. front++ (front=54). 队列: [c]</p><p>110. 进队操作: 元素d入队. rear++ (rear=55). 队列: [c, d]</p><p>111. 出队操作: 元素c出队. front++ (front=55). 队列: [d]</p><p>112. 进队操作: 元素e入队. rear++ (rear=56). 队列: [d, e]</p><p>113. 出队操作: 元素d出队. front++ (front=56). 队列: [e]</p><p>114. 进队操作: 元素f入队. rear++ (rear=57). 队列: [e, f]</p><p>115. 出队操作: 元素e出队. front++ (front=57). 队列: [f]</p><p>116. 进队操作: 元素g入队. rear++ (rear=58). 队列: [f, g]</p><p>117. 出队操作: 元素f出队. front++ (front=58). 队列: [g]</p><p>118. 进队操作: 元素h入队. rear++ (rear=59). 队列: [g, h]</p><p>119. 出队操作: 元素g出队. front++ (front=59). 队列: [h]</p><p>120. 进队操作: 元素i入队. rear++ (rear=60). 队列: [h, i]</p><p>121. 出队操作: 元素h出队. front++ (front=60). 队列: [i]</p><p>122. 进队操作: 元素j入队. rear++ (rear=61). 队列: [i, j]</p><p>123. 出队操作: 元素i出队. front++ (front=61). 队列: [j]</p><p>124. 进队操作: 元素k入队. rear++ (rear=62). 队列: [j, k]</p><p>125. 出队操作: 元素j出队. front++ (front=62). 队列: [k]</p><p>126. 进队操作: 元素l入队. rear++ (rear=63). 队列: [k, l]</p><p>127. 出队操作: 元素k出队. front++ (front=63). 队列: [l]</p><p>128. 进队操作: 元素m入队. rear++ (rear=64). 队列: [l, m]</p><p>129. 出队操作: 元素l出队. front++ (front=64). 队列: [m]</p><p>130. 进队操作: 元素n入队. rear++ (rear=65). 队列: [m, n]</p><p>131. 出队操作: 元素m出队. front++ (front=65). 队列: [n]</p><p>132. 进队操作: 元素o入队. rear++ (rear=66). 队列: [n, o]</p><p>133. 出队操作: 元素n出队. front++ (front=66). 队列: [o]</p><p>134. 进队操作: 元素p入队. rear++ (rear=67). 队列: [o, p]</p><p>135. 出队操作: 元素o出队. front++ (front=67). 队列: [p]</p><p>136. 进队操作: 元素q入队. rear++ (rear=68). 队列: [p, q]</p><p>137. 出队操作: 元素p出队. front++ (front=68). 队列: [q]</p><p>138. 进队操作: 元素r入队. rear++ (rear=69). 队列: [q, r]</p><p>139. 出队操作: 元素q出队. front++ (front=69). 队列: [r]</p><p>140. 进队操作: 元素s入队. rear++ (rear=70). 队列: [r, s]</p><p>141. 出队操作: 元素r出队. front++ (front=70). 队列: [s]</p><p>142. 进队操作: 元素t入队. rear++ (rear=71). 队列: [s, t]</p><p>143. 出队操作: 元素s出队. front++ (front=71). 队列: [t]</p><p>144. 进队操作: 元素u入队. rear++ (rear=72). 队列: [t, u]</p><p>145. 出队操作: 元素t出队. front++ (front=72). 队列: [u]</p><p>146. 进队操作: 元素v入队. rear++ (rear=73). 队列: [u, v]</p><p>147. 出队操作: 元素u出队. front++ (front=73). 队列: [v]</p><p>148. 进队操作: 元素w入队. rear++ (rear=74). 队列: [v, w]</p><p>149. 出队操作: 元素v出队. front++ (front=74). 队列: [w]</p><p>150. 进队操作: 元素x入队. rear++ (rear=75). 队列: [w, x]</p><p>151. 出队操作: 元素w出队. front++ (front=75). 队列: [x]</p><p>152. 进队操作: 元素y入队. rear++ (rear=76). 队列: [x, y]</p><p>153. 出队操作: 元素x出队. front++ (front=76). 队列: [y]</p><p>154. 进队操作: 元素z入队. rear++ (rear=77). 队列: [y, z]</p><p>155. 出队操作: 元素y出队. front++ (front=77). 队列: [z]</p><p>156. 进队操作: 元素a入队. rear++ (rear=78). 队列: [z, a]</p><p>157. 出队操作: 元素z出队. front++ (front=78). 队列: [a]</p><p>158. 进队操作: 元素b入队. rear++ (rear=79). 队列: [a, b]</p><p>159. 出队操作: 元素a出队. front++ (front=79). 队列: [b]</p><p>160. 进队操作: 元素c入队. rear++ (rear=80). 队列: [b, c]</p><p>161. 出队操作: 元素b出队. front++ (front=80). 队列: [c]</p><p>162. 进队操作: 元素d入队. rear++ (rear=81). 队列: [c, d]</p><p>163. 出队操作: 元素c出队. front++ (front=81). 队列: [d]</p><p>164. 进队操作: 元素e入队. rear++ (rear=82). 队列: [d, e]</p><p>165. 出队操作: 元素d出队. front++ (front=82). 队列: [e]</p><p>166. 进队操作: 元素f入队. rear++ (rear=83). 队列: [e, f]</p><p>167. 出队操作: 元素e出队. front++ (front=83). 队列: [f]</p><p>168. 进队操作: 元素g入队. rear++ (rear=84). 队列: [f, g]</p><p>169. 出队操作: 元素f出队. front++ (front=84). 队列: [g]</p><p>170. 进队操作: 元素h入队. rear++ (rear=85). 队列: [g, h]</p><p>171. 出队操作: 元素g出队. front++ (front=85). 队列: [h]</p><p>172. 进队操作: 元素i入队. rear++ (rear=86). 队列: [h, i]</p><p>173. 出队操作: 元素h出队. front++ (front=86). 队列: [i]</p><p>174. 进队操作: 元素j入队. rear++ (rear=87). 队列: [i, j]</p><p>175. 出队操作: 元素i出队. front++ (front=87). 队列: [j]</p><p>176. 进队操作: 元素k入队. rear++ (rear=88). 队列: [j, k]</p><p>177. 出队操作: 元素j出队. front++ (front=88). 队列: [k]</p><p>178. 进队操作: 元素l入队. rear++ (rear=89). 队列: [k, l]</p><p>179. 出队操作: 元素k出队. front++ (front=89). 队列: [l]</p><p>180. 进队操作: 元素m入队. rear++ (rear=90). 队列: [l, m]</p><p>181. 出队操作: 元素l出队. front++ (front=90). 队列: [m]</p><p>182. 进队操作: 元素n入队. rear++ (rear=91). 队列: [m, n]</p><p>183. 出队操作: 元素m出队. front++ (front=91). 队列: [n]</p><p>184. 进队操作: 元素o入队. rear++ (rear=92). 队列: [n, o]</p><p>185. 出队操作: 元素n出队. front++ (front=92). 队列: [o]</p><p>186. 进队操作: 元素p入队. rear++ (rear=93). 队列: [o, p]</p><p>187. 出队操作: 元素o出队. front++ (front=93). 队列: [p]</p><p>188. 进队操作: 元素q入队. rear++ (rear=94). 队列: [p, q]</p><p>189. 出队操作: 元素p出队. front++ (front=94). 队列: [q]</p><p>190. 进队操作: 元素r入队. rear++ (rear=95). 队列: [q, r]</p><p>191. 出队操作: 元素q出队. front++ (front=95). 队列: [r]</p><p>192. 进队操作: 元素s入队. rear++ (rear=96). 队列: [r, s]</p><p>193. 出队操作: 元素r出队. front++ (front=96). 队列: [s]</p><p>194. 进队操作: 元素t入队. rear++ (rear=97). 队列: [s, t]</p><p>195. 出队操作: 元素s出队. front++ (front=97). 队列: [t]</p><p>196. 进队操作: 元素u入队. rear++ (rear=98). 队列: [t, u]</p><p>197. 出队操作: 元素t出队. front++ (front=98). 队列: [u]</p><p>198. 进队操作: 元素v入队. rear++ (rear=99). 队列: [u, v]</p><p>199. 出队操作: 元素u出队. front++ (front=99). 队列: [v]</p><p>200. 进队操作: 元素w入队. rear++ (rear=100). 队列: [v, w]</p><p>201. 出队操作: 元素v出队. front++ (front=100). 队列: [w]</p><p>202. 进队操作: 元素x入队. rear++ (rear=101). 队列: [w, x]</p><p>203. 出队操作: 元素w出队. front++ (front=101). 队列: [x]</p><p>204. 进队操作: 元素y入队. rear++ (rear=102). 队列: [x, y]</p><p>205. 出队操作: 元素x出队. front++ (front=102). 队列: [y]</p><p>206. 进队操作: 元素z入队. rear++ (rear=103). 队列: [y, z]</p><p>207. 出队操作: 元素y出队. front++ (front=103). 队列: [z]</p><p>208. 进队操作: 元素a入队. rear++ (rear=104). 队列: [z, a]</p><p>209. 出队操作: 元素z出队. front++ (front=104). 队列: [a]</p><p>210. 进队操作: 元素b入队. rear++ (rear=105). 队列: [a, b]</p><p>211. 出队操作: 元素a出队. front++ (front=105). 队列: [b]</p><p>212. 进队操作: 元素c入队. rear++ (rear=106). 队列: [b, c]</p><p>213. 出队操作: 元素b出队. front++ (front=106). 队列: [c]</p><p>214. 进队操作: 元素d入队. rear++ (rear=107). 队列: [c, d]</p><p>215. 出队操作: 元素c出队. front++ (front=107). 队列: [d]</p><p>216. 进队操作: 元素e入队. rear++ (rear=108). 队列: [d, e]</p><p>217. 出队操作: 元素d出队. front++ (front=108). 队列: [e]</p><p>218. 进队操作: 元素f入队. rear++ (rear=109). 队列: [e, f]</p><p>219. 出队操作: 元素e出队. front++ (front=109). 队列: [f]</p><p>220. 进队操作: 元素g入队. rear++ (rear=110). 队列: [f, g]</p><p>221. 出队操作: 元素f出队. front++ (front=110). 队列: [g]</p><p>222. 进队操作: 元素h入队. rear++ (rear=111). 队列: [g, h]</p><p>223. 出队操作: 元素g出队. front++ (front=111). 队列: [h]</p><p>224. 进队操作: 元素i入队. rear++ (rear=112). 队列: [h, i]</p><p>225. 出队操作: 元素h出队. front++ (front=112). 队列: [i]</p><p>226. 进队操作: 元素j入队. rear++ (rear=113). 队列: [i, j]</p><p>227. 出队操作: 元素i出队. front++ (front=113). 队列: [j]</p><p>228. 进队操作: 元素k入队. rear++ (rear=114). 队列: [j, k]</p><p>229. 出队操作: 元素j出队. front++ (front=114). 队列: [k]</p><p>230. 进队操作: 元素l入队. rear++ (rear=115). 队列: [k, l]</p><p>231. 出队操作: 元素k出队. front++ (front=115). 队列: [l]</p><p>232. 进队操作: 元素m入队. rear++ (rear=116). 队列: [l, m]</p><p>233. 出队操作: 元素l出队. front++ (front=116). 队列: [m]</p><p>234. 进队操作: 元素n入队. rear++ (rear=117). 队列: [m, n]</p><p>235. 出队操作: 元素m出队. front++ (front=117). 队列: [n]</p><p>236. 进队操作: 元素o入队. rear++ (rear=118). 队列: [n, o]</p><p>237. 出队操作: 元素n出队. front++ (front=118). 队列: [o]</p><p>238. 进队操作: 元素p入队. rear++ (rear=119). 队列: [o, p]</p><p>239. 出队操作: 元素o出队. front++ (front=119). 队列: [p]</p><p>240. 进队操作: 元素q入队. rear++ (rear=120). 队列: [p, q]</p><p>241. 出队操作: 元素p出队. front++ (front=120). 队列: [q]</p><p>242. 进队操作: 元素r入队. rear++ (rear=121). 队列: [q, r]</p><p>243. 出队操作: 元素q出队. front++ (front=121). 队列: [r]</p><p>244. 进队操作: 元素s入队. rear++ (rear=122). 队列: [r, s]</p><p>245. 出队操作: 元素r出队. front++ (front=122). 队列: [s]</p><p>246. 进队操作: 元素t入队. rear++ (rear=123). 队列: [s, t]</p><p>247. 出队操作: 元素s出队. front++ (front=123). 队列: [t]</p><p>248. 进队操作: 元素u入队. rear++ (rear=124). 队列: [t, u]</p><p>249. 出队操作: 元素t出队. front++ (front=124). 队列: [u]</p><p>250. 进队操作: 元素v入队. rear++ (rear=125). 队列: [u, v]</p><p>251. 出队操作: 元素u出队. front++ (front=125). 队列: [v]</p><p>252. 进队操作: 元素w入队. rear++ (rear=126). 队列: [v, w]</p><p>253. 出队操作: 元素v出队. front++ (front=126). 队列: [w]</p><p>254. 进队操作: 元素x入队. rear++ (rear=127). 队列: [w, x]</p><p>255. 出队操作: 元素w出队. front++ (front=127). 队列: [x]</p><p>256. 进队操作: 元素y入队. rear++ (rear=128). 队列: [x, y]</p><p>257. 出队操作: 元素x出队. front++ (front=128). 队列: [y]</p><p>258. 进队操作: 元素z入队. rear++ (rear=129). 队列: [y, z]</p><p>259. 出队操作: 元素y出队. front++ (front=129). 队列: [z]</p><p>260. 进队操作: 元素a入队. rear++ (rear=130). 队列: [z, a]</p><p>261. 出队操作: 元素z出队. front++ (front=130). 队列: [a]</p><p>262. 进队操作: 元素b入队. rear++ (rear=131). 队列: [a, b]</p><p>263. 出队操作: 元素a出队. front++ (front=131). 队列: [b]</p><p>264. 进队操作: 元素c入队. rear++ (rear=132). 队列: [b, c]</p><p>265. 出队操作: 元素b出队. front++ (front=132). 队列: [c]</p><p>266. 进队操作: 元素d入队. rear++ (rear=133). 队列: [c, d]</p><p>267. 出队操作: 元素c出队. front++ (front=133). 队列: [d]</p><p>268. 进队操作: 元素e入队. rear++ (rear=134). 队列: [d, e]</p><p>269. 出队操作: 元素d出队. front++ (front=134). 队列: [e]</p><p>270. 进队操作: 元素f入队. rear++ (rear=135). 队列: [e, f]</p><p>271. 出队操作: 元素e出队. front++ (front=135). 队列: [f]</p><p>272. 进队操作: 元素g入队. rear++ (rear=136). 队列: [f, g]</p><p>273. 出队操作: 元素f出队. front++ (front=136). 队列: [g]</p><p>274. 进队操作: 元素h入队. rear++ (rear=137). 队列: [g, h]</p><p>275. 出队操作: 元素g出队. front++ (front=137). 队列: [h]</p><p>276. 进队操作: 元素i入队. rear++ (rear=138). 队列: [h, i]</p><p>277. 出队操作: 元素h出队. front++ (front=138). 队列: [i]</p><p>278. 进队操作: 元素j入队. rear++ (rear=139). 队列: [i, j]</p><p>279. 出队操作: 元素i出队. front++ (front=139). 队列: [j]</p><p>280. 进队操作: 元素k入队. rear++ (rear=140). 队列: [j, k]</p><p>281. 出队操作: 元素j出队. front++ (front=140). 队列: [k]</p><p>282. 进队操作: 元素l入队. rear++ (rear=141). 队列: [k, l]</p><p>283. 出队操作: 元素k出队. front++ (front=141). 队列: [l]</p><p>284. 进队操作: 元素m入队. rear++ (rear=142). 队列: [l, m]</p><p>285. 出队操作: 元素l出队. front++ (front=142). 队列: [m]</p><p>286. 进队操作: 元素n入队. rear++ (rear=143). 队列: [m, n]</p><p>287. 出队操作: 元素m出队. front++ (front=143). 队列: [n]</p><p>288. 进队操作: 元素o入队. rear++ (rear=144). 队列: [n, o]</p><p>289. 出队操作: 元素n出队. front++ (front=144). 队列: [o]</p><p>290. 进队操作: 元素p入队. rear++ (rear=145). 队列: [o, p]</p><p>291. 出队操作: 元素o出队. front++ (front=145). 队列: [p]</p><p>292. 进队操作: 元素q入队. rear++ (rear=146). 队列: [p, q]</p><p>293. 出队操作: 元素p出队. front++ (front=146). 队列: [q]</p><p>294. 进队操作: 元素r入队. rear++ (rear=147). 队列: [q, r]</p><p>295. 出队操作: 元素q出队. front++ (front=147). 队列: [r]</p><p>296. 进队操作: 元素s入队. rear++ (rear=148). 队列: [r, s]</p><p>297. 出队操作: 元素r出队. front++ (front=148). 队列: [s]</p><p>298. 进队操作: 元素t入队. rear++ (rear=149). 队列: [s, t]</p><p>299. 出队操作: 元素s出队. front++ (front=149). 队列: [t]</p><p>300. 进队操作: 元素u入队. rear++ (rear=150). 队列: [t, u]</p><p>301. 出队操作: 元素t出队. front++ (front=150). 队列: [u]</p><p>302. 进队操作: 元素v入队. rear++ (rear=151). 队列: [u, v]</p><p>303. 出队操作: 元素u出队. front++ (front=151). 队列: [v]</p><p>304. 进队操作: 元素w入队. rear++ (rear=152). 队列: [v, w]</p><p>305. 出队操作: 元素v出队. front++ (front=152). 队列: [w]</p><p>306. 进队操作: 元素x入队. rear++ (rear=153). 队列: [w, x]</p><p>307. 出队操作: 元素w出队. front++ (front=153). 队列: [x]</p><p>308. 进队操作: 元素y入队. rear++ (rear=154). 队列: [x, y]</p><p>309. 出队操作: 元素x出队. front++ (front=154). 队列: [y]</p><p>310. 进队操作: 元素z入队. rear++ (rear=155). 队列: [y, z]</p><p>311. 出队操作: 元素y出队. front++ (front=155). 队列: [z]</p><p>312. 进队操作: 元素a入队. rear++ (rear=156). 队列: [z, a]</p><p>313. 出队操作: 元素z出队. front++ (front=156). 队列: [a]</p><p>314. 进队操作: 元素b入队. rear++ (rear=157). 队列: [a, b]</p><p>315. 出队操作: 元素a出队. front++ (front=157). 队列: [b]</p><p>316. 进队操作: 元素c入队. rear++ (rear=158). 队列: [b, c]</p><p>317. 出队操作: 元素b出队. front++ (front=158). 队列: [c]</p><p>318. 进队操作: 元素d入队. rear++ (rear=159). 队列: [c, d]</p><p>319. 出队操作: 元素c出队. front++ (front=159). 队列: [d]</p><p>320. 进队操作: 元素e入队. rear++ (rear=160). 队列: [d, e]</p><p>321. 出队操作: 元素d出队. front++ (front=160). 队列: [e]</p><p>322. 进队操作: 元素f入队. rear++ (rear=161). 队列: [e, f]</p><p>323. 出队操作: 元素e出队. front++ (front=161). 队列: [f]</p><p>324. 进队操作: 元素g入队. rear++ (rear=162). 队列: [f, g]</p><p>325. 出队操作: 元素f出队. front++ (front=162). 队列: [g]</p><p>326. 进队操作: 元素h入队. rear++ (rear=163). 队列: [g, h]</p><p>327. 出队操作: 元素g出队. front++ (front=163). 队列: [h]</p><p>328. 进队操作: 元素i入队. rear++ (rear=164). 队列: [h, i]</p><p>329. 出队操作: 元素h出队. front++ (front=164). 队列: [i]</p><p>330. 进队操作: 元素j入队. rear++ (rear=165). 队列: [i, j]</p><p>331. 出队操作: 元素i出队. front++ (front=165). 队列: [j]</p><p>332. 进队操作: 元素k入队. rear++ (rear=166). 队列: [j, k]</p><p>333. 出队操作: 元素j出队. front++ (front=166). 队列: [k]</p><p>334. 进队操作: 元素l入队. rear++ (rear=167). 队列: [k, l]</p><p>335. 出队操作: 元素k出队. front++ (front=167). 队列: [l]</p><p>336. 进队操作: 元素m入队. rear++ (rear=168). 队列: [l, m]</p><p>337. 出队操作: 元素l出队. front++ (front=168). 队列: [m]</p><p>338. 进队操作: 元素n入队. rear++ (rear=169). 队列: [m, n]</p><p>339. 出队操作: 元素m出队. front++ (front=169). 队列: [n]</p><p>340. 进队操作: 元素o入队. rear++ (rear=170). 队列: [n, o]</p><p>341. 出队操作: 元素n出队. front++ (front=170). 队列: [o]</p><p>342. 进队操作: 元素p入队. rear++ (rear=171). 队列: [o, p]</p><p>343. 出队操作: 元素o出队. front++ (front=171). 队列: [p]</p><p>344. 进队操作: 元素q入队. rear++ (rear=172). 队列: [p, q]</p><p>345. 出队操作: 元素p出队. front++ (front=172). 队列: [q]</p><p>346. 进队操作: 元素r入队. rear++ (rear=173). 队列: [q, r]</p><p>347. 出队操作: 元素q出队. front++ (front=173). 队列: [r]</p><p>348. 进队操作: 元素s入队. rear++ (rear=174). 队列: [r, s]</p><p>349. 出队操作: 元素r出队. front++ (front=174).</p></div>											