

数据结构第一次作业

邵志豪

March 7, 2023

1.8 试确定各程序段中 @ 标记语句的频度：

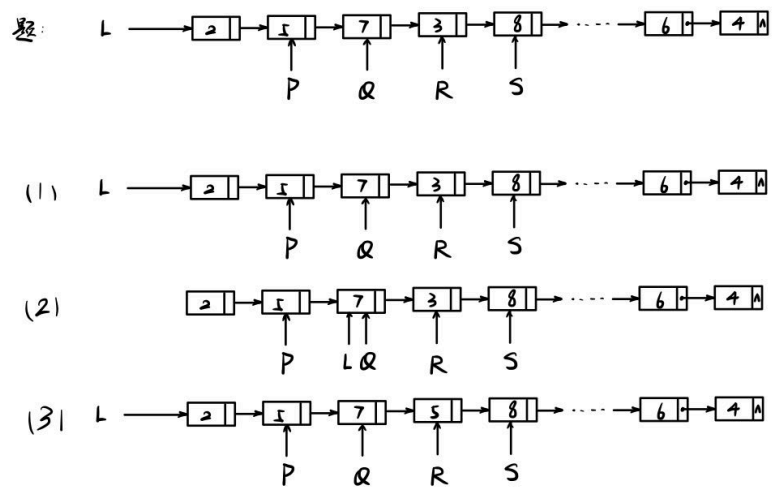
- (1) $n - 1$
- (2) $n - 1$
- (3) $n - 1$
- (4) $\frac{(n+1)n}{2}$
- (5) $\frac{n(n+1)(n+2)}{6}$
- (6) n
- (7) $\lfloor \sqrt{n} \rfloor$
- (8) 1100

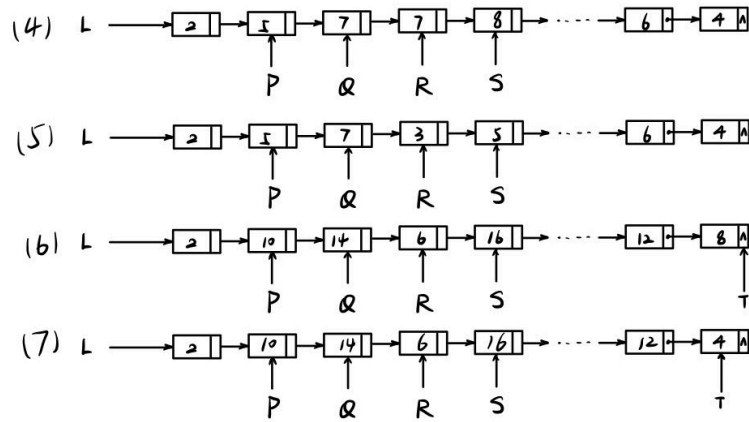
1.9 $count = \lceil \log_2 n \rceil - 2$, $T(n) = O(\log n)$

1.12 判断是否正确：

- (1) 正确
- (2) 正确
- (3) 错误
- (4) 正确
- (5) 错误

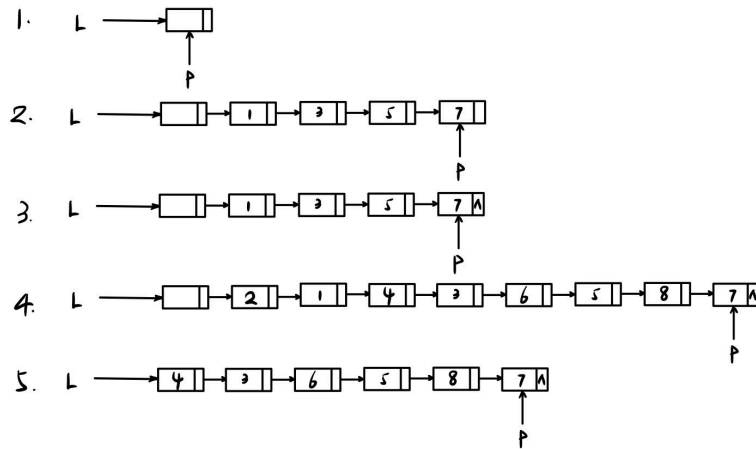
2.4 原状态及结果如下图：





2.5 各行运行结果如下图:

2.5



2.9 简述以下算法功能:

- (1) 将多数数据元素单链表 L 的第一个数据元素和第二个数据元素接在最后一个数据元素之后, 并和第二个数据元素解开链接。
- (2) 将单循环链表以 pa 和 pb 指向的数据元素之前为标记, 截断成两段, 并各自首尾相连, 成为两个单循环链表。

3.3 stack

3.7 变化过程如下：

步骤	OPTR 栈	OPND 栈	输入字符	主要操作
1	#		<u>A</u> -B×C/D+E↑F#	PUSH(OPND,A)
2	#	A	<u> </u> -B×C/D+E↑F#	PUSH(OPTR,'-')
3	# -	A	<u>B</u> ×C/D+E↑F#	PUSH(OPND,B)
4	# -	A B	<u> </u> ×C/D+E↑F#	PUSH(OPTR,'×')
5	# - ×	A B	<u>C</u> /D+E↑F#	PUSH(OPND,C)
6	# - ×	A B C	<u> </u> /D+E↑F#	operate(B,'×',C)
7	# - ×	A B×C	<u> </u> /D+E↑F#	PUSH(OPTR,'/')
8	# - /	A B×C	<u>D</u> +E↑F#	PUSH(OPND,D)
9	# - /	A B×C D	<u> </u> +E↑F#	operate(B×C,'/',D)
10	# -	A B×C/D	<u> </u> +E↑F#	operate(A,'-',B×C/D)
11	#	A-B×C/D	<u> </u> +E↑F#	PUSH(OPTR,'+')
12	# +	A-B×C/D	<u>E</u> ↑F#	PUSH(OPND,E)
13	# +	A-B×C/D E	<u> </u> ↑F#	PUSH(OPTR,'↑')
14	# + ↑	A-B×C/D E	<u>F</u> #	PUSH(OPND,F)
15	# + ↑	A-B×C/D E F	#	operate(E,'↑',F)
16	# +	A-B×C/D E↑F	#	operate(A-B×C/D,'+',E↑F)
17	#	A-B×C/D+E↑F	#	RETURN(GETTOP(OPND))

3.11 队列和栈都是操作受限的线性表，他们的数据结构都是线性表，但是他们首先得操作不同：前者要求先进先出，后者要求先进后出；前者要求在线性表的一端插入，另一端删除，后者则只在一端插入和删除；前者遍历时不会影响数据结构，也不会另外开辟空间，后者则需要弹出所需数据之上的所有数据才能遍历，而且需要额外空间进行暂时存储。