

```

Pop(s, x);
sum += x;
cout << sum << endl;
}
DestroyStack(s);
}

```

第4章

4.3

15, 5, "STUDENT", "0"
 6 Null "I AM A WORKER"
 A GOOD STUDENT

4.4

s = "THIS SAMPLE IS" t = "A GOOD"
 u = "ONE" v = "THIS SAMPLE IS A GOOD O"
 StrLength(s) = 15 Index(v, y) = 2 Index(u, y) =

4.8

搜索词 ADABBA DADA
 匹配值 1010012323
 AD¹BA¹DA¹BB¹AA¹BA¹DA¹BB¹DA¹DADA
 A¹DA¹BB¹AD¹ADA¹

1.8.

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

1.9.

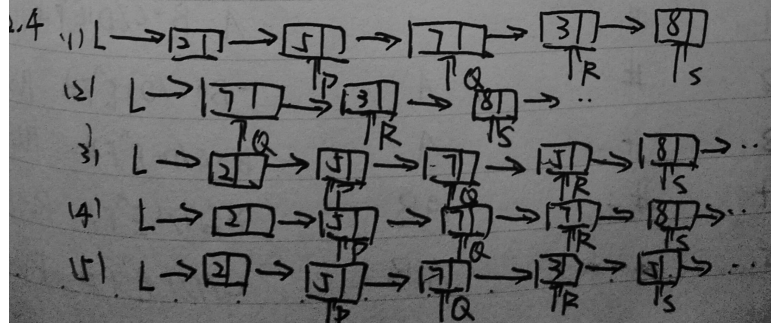
$$O(\log_2 n) \quad \text{count} = \log_2 n - 2$$

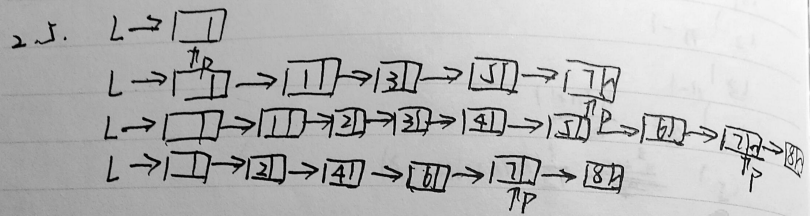
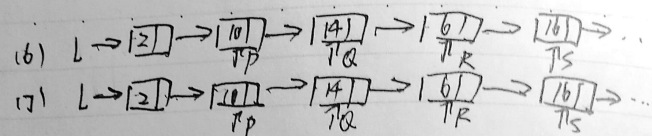
1.12.

이 문제는 3가지 경우로 나뉜다

1.13.

2.4.





2.9. (1) 若 L 的长度不小于 2, 则首元结点变为尾元结点

(2) 将一个单循环链表拆成两个单循环链表

第三章

3.3.

stack

3.7

$$A - B * C / D + E * F$$

$$B * C = G$$

$$G / D = H \quad A - H = I$$

$$E * F = J \quad I + J = K$$

步骤

OPTR 栈

OPND 栈

输入字符 主要操作

1

#

A

$A - B * C / D + E * F \#$ PUSH(OPND, A)

2

#

A

$- B * C / D + E * F \#$ PUSH(OPTR, -)

3

#

AB

$B * C / D + E * F \#$ PUSH(OPND, B)

4

#

AB

$* C / D + E * F \#$ PUSH(OPTR, *)

5

#

AB

$C / D + E * F \#$ PUSH(OPND, C)

6	# - *	A B C	/D + E^F #	Operate (B, *, C),
7	# - .	A G	/D + E^F #	PUSH (OPTR, /)
8	# - /	A G	D + E^F #	PUSH (OPND, D)
9	# - /	A G D	+ E^F #	Operate (G, /, D)
10	# -	A H	+ E^F #	Operate (A, -, H)
11	#	I	+ E^F #	PUSH (OPTR, +)
12	# +	I	E^F #	PUSH (OPND, E)
13	# +	I E	^F #	PUSH (OPTR, ^)
14	# + ^	I E	F #	PUSH (OPND, F)
15	# + ^	I E F	#	Operate (E, ^, F)
16	# +	I J	#	Operate (I, +, J)
17	#	K	#	Return

```
void test (int &sum)
```

```
{
```

```
Stack s;
```

```
InitStack (s);
```

```
int x;
```

```
do {
```

```
cin >> x;
```

```
Push (s, x);
```

```
} while (x > 0);
```

```
while (!StackEmpty(s)) {
```