

郑 州 西 亚 斯 学 院

学 生 作 业 纸

SIAS UNIVERSITY EXERCISE NOTES

姓名 NAME

学号 STUDENTID

班级 CLASS

一. 线性表

1.1 顺序表

(1) 初始化

```
L->data = new Sqlist;
```

```
if (!L->data) exit(10);
```

```
L->length = 0;
```

(2) 取值

```
if (i < 1 || i > L->length) return;
```

```
e = L->data[i-1];
```

(3) 查找

```
for (i = 0; i < L->length; i++)
```

```
if (L->data[i] == e) return i;
```

(4) 插入

```
if (!IsValidValue(i)) return
```

```
if (L->length == MAXSIZE) return;
```

```
for (j = L->length-1; j >= i-1; j++)
```

```
L->data[j+1] = L->data[j];
```

```
L->data[i-1] = e;
```

(5) 删除

```
for (j = i; j < L->length-1; j++)
```

```
L->data[j-1] = L->data[j]
```

```
L->length--;
```

郑州西亚斯学院

学生作业纸

SIAS UNIVERSITY EXERCISE NOTES

姓名 NAME

学号 STUDENTID

班级 CLASS

1.2 链表

1) 初始化

$L = \text{new Linklist};$

$\text{if} ((L) \text{ exit}(0));$

$L \rightarrow \text{next} = \text{NULL};$

14) 遍历

$\text{for} (j=0; j<i; j++)$

$L = L \rightarrow \text{next};$

2) 尾插法建表

$\text{Linklist } L = \text{InitList}();$

$\text{Node } *r = L;$

$\text{for} (i=0; i<n; i++)$

$\{ p = \text{new Node};$

$p \rightarrow \text{data} = e;$

$p \rightarrow \text{next} = \text{NULL};$

$r \rightarrow \text{next} = p;$

$r = p; \}$

15) 删除

$\text{for} (\text{int } j=0; j<i-1; j++)$

$p = p \rightarrow \text{next};$

$q = p \rightarrow \text{next};$

$p \rightarrow \text{next} = p \rightarrow \text{next} \rightarrow \text{next};$

$\text{free}(q);$

13) 插入

$\text{Node } *pre, *p$

$\text{for} (j=1; j<i; j++)$

$pre = pre \rightarrow \text{next};$

$p = \text{new Node};$

$p \rightarrow \text{data} = e; p \rightarrow \text{next} = pre \rightarrow \text{next}; pre \rightarrow \text{next} = p$

郑州西亚斯学院

学生作业纸

SIAS UNIVERSITY EXERCISE NOTES

姓名NAME

学号STUDENTID

班级CLASS

1.3. 双向链表

1) 插入

$s = \text{new DuLNode};$

$s \rightarrow \text{data} = e$

$s \rightarrow \text{prior} = p \rightarrow \text{prior};$

$p \rightarrow \text{prior} \rightarrow \text{next} = s;$

$s \rightarrow \text{next} = p;$

$p \rightarrow \text{prior} = s;$

2) 删除

$p \rightarrow \text{prior} \rightarrow \text{next} = p \rightarrow \text{next};$

$p \rightarrow \text{next} \rightarrow \text{prior} = p \rightarrow \text{prior};$

$\text{free}(p);$

二. 栈和队列

1. 顺序栈

1) 初始化

$s \rightarrow \text{base} = \text{new SqStack};$

$\text{if}(s \rightarrow \text{base} == \text{NULL}) \text{exit}(1);$

$s \rightarrow \text{top} = s \rightarrow \text{base};$

$s \rightarrow \text{stackSize} = \text{MAXSIZE};$

2) 入栈

$\text{if}(s \rightarrow \text{top} - s \rightarrow \text{base} == s \rightarrow \text{stackSize}) \text{return}$

$s \rightarrow \text{top} ++ = e;$

3) 出栈

$\text{if}(s \rightarrow \text{top} = s \rightarrow \text{base}) \text{return};$

$e = *--s \rightarrow \text{top};$

郑州西亚斯学院

学生作业纸

SIAS UNIVERSITY EXERCISE NOTES

姓名NAME _____ 学号STUDENTID _____ 班级CLASS _____

1. 和栈顶元素

if (s.top != s.base) return *(s.top - 1);

2. 链栈

1) 初始化

S = NULL

2) 入栈

p = new Linkstack;

p->data = e;

p->next = S;

S = p;

3) 出栈

if (S == NULL) return;

e = S->data;

p = S;

S = S->next;

free(p);

4) 取栈顶元素

if (S != NULL) return

return S->data;

郑州西亚斯学院

学生作业纸

SIAS UNIVERSITY EXERCISE NOTES

姓名NAME

学号STUDENTID

班级CLASS

3. 队列

3.1 顺序队列

1) 初始化

$p \rightarrow \text{front} = p \rightarrow \text{rear} = 0;$

2) 入队

$p \rightarrow \text{rear}++;$

$p \rightarrow Q[p \rightarrow \text{rear}] = e;$

3) 出队

$p \rightarrow \text{front}++;$

$e = p \rightarrow Q[p \rightarrow \text{front}];$

3.2 链式队列

1) 初始化

$Q \rightarrow \text{base} = \text{new SqQueue};$

$\text{if} (!Q \rightarrow \text{base}) \text{exit}(10);$

$Q \rightarrow \text{front} = Q \rightarrow \text{rear} = 0;$

2) 求队长

$\text{return} (Q \rightarrow \text{rear} - Q \rightarrow \text{front} + \text{MAXQSIZE}) \% \text{MAXQSIZE};$

3) 入队

$\text{if} ((Q \rightarrow \text{rear} + 1) \% M == Q \rightarrow \text{front})$

郑 州 西 亚 斯 学 院

学 生 作 业 纸

SIAS UNIVERSITY EXERCISE NOTES

姓名NAME _____ 学号STUDENTID _____ 班级CLASS _____

4.1 出队

if (Q.front == Q.rear) return;

e = Q.base[Q.front];

Q.front = (Q.front + 1) % M;

4.2 取队头元素

if (Q.front != Q.rear)

return Q.base[Q.front];

3.3 链队

1.1 初始化

Q.front = Q.rear = new LinkQueue;

Q.front->next = NULL;

1.2 入队

p = new QNode;

p->data = e;

p->next = NULL;

Q.rear->next = p;

Q.rear = p;

1.3 出队

p = Q.front->next;

if (Q.rear == p) Q.rear = Q.front;

e = p->data;

free(p);

Q.front->next = p->next;

郑州西亚斯学院

学生作业纸

SIAS UNIVERSITY EXERCISE NOTES

姓名NAME _____ 学号STUDENTID _____ 班级CLASS _____

4) 初始化函数。

```
if (Q.front != Q.rear)
```

```
return Q.front ->next ->data;
```

三. 串和数组。

1. BF 算法。

```
i = pos; j = 1
```

```
while (i <= S[L0] && j <= T[L0]) {
```

```
    if (S[i] == T[j]) { i++; j++; }
```

```
    else { i = i - j + 2; j = 1; }
```

```
    if (j > T[L0]) return i - T[L0];
```

```
    else return 0;
```

3. next 函数。

```
i = 1; next[L1] = 0; j = 0;
```

```
while (i <= T[L0]) {
```

```
    if (j == 0 || T[i] == T[j]) {
```

```
        i++; j++; next[L1] = j;
```

```
    } else j = next[j]; }
```

2. KMP 算法。

```
i = pos; j = 1;
```

```
while (i <= S[L0] && j <= T[L0]) {
```

```
    if (j == 0 || S[i] == T[j]) { i++; j++; }
```

```
    else j = next[j]; }
```

```
if (j > T[L0]) return i - T[L0];
```

```
else return 0;
```