苏州大学 数据结构 课程3卷参考答案（共 3 页）

院系 专业 \_\_\_\_\_\_\_\_\_\_

1. 填空（2分×15）
2. front=rear；(rear+2) % n = front
3. O（n1/2）

3、AB+CD-/EF-TR-\*+

4、5 ，空

5、4265713；4675231

6、h，2h-1

7、各边权值之和最小的生成树，prim或kruscal

8、add\_to.c\_str(); add\_on.c\_str(); add\_to=copy;

1. 应用题

1、ABCDEFGH，ABDCEFHG （各4分）

2、 （1）ASLsucc=31/12，ASLunsucc=60/14 （6分）

（2）ASLsucc=18/12，ASLunsucc=12/14 （6分）

3、初始关键字：68,5,16,94,23,71,73,72 ,69

i=2：5,68, 16,94,23,71,73,72 ,69

i=3：5, 16,68, 94,23,71,73,72 ,69

i=4：5, 16,68, 94,23,71,73,72 ,69

i=5：5, 16, 23,68, 94, 71,73,72 ,69

i=6：5, 16, 23,68, 71,94, 73,72 ,69

i=7：5, 16, 23,68, 71, 73,94, 72 ,69

i=8：5, 16, 23,68, 71, 72 ,73,94, 69

i=9：5, 16, 23,68, 69,71, 72 ,73,94 （7分）

当把69插入有序表时，所需的比较次数和移动次数分别为5次和6次。 （2分）

算法需要一个记录的辅助空间。 （1分）

4、

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 终点 | i=1 | i=2 | i=3 | i=4 | i=5 | i=6 |
| 2 | 45  (1,2) | 45  (1,2) | 40  (1,5,2) | 40  (1,5,2) |  |  |
| 3 | ∝ | ∝ | 55  (1,5,3) | 55  (1,5,3) | 52  (1,5,2,3) | 50  (1,4,6,3) |
| 4 | 12  (1,4) |  |  |  |  |  |
| 5 | 25  (1,5) | 25  (1,5) |  |  |  |  |
| 6 | ∝ | 42  (1,4,6) | 42  (1,4,6) | 42  (1,4,6) | 42  (1,4,6) |  |
| 7 | 35  (1,7) | 32  (1,4,7) | 32  (1,4,7) |  |  |  |
| 选择v | 4 | 5 | 7 | 2 | 6 | 3 |
| S | {1,4} | {1,4,5} | {1,4,5,7} | {1,4,5,7,2} | {1,4,5,7,2,6} | {1,4,5,7,2,6,3} |

1. 算法设计题

template <class List\_entry>

Error\_code List<List\_entry>::remove(int position, const List\_entry &x)

status del(LinkList L,int i) {

if (position<=0) return ERROR;

return OK;

if (position==1) {

s=head;

head=head->next;

free(s);

}

else {

j=1; s=head;

while ((j<positon) && (s!=null)

{

p=s;s=s->next;

j++;

}

if (s=null) return ERROR;

else {

p->next=s->next;

free(s);

}

}

}

1. （1）

tpedef struct {

SelemType \*base[1];

SelemType \*top[1];

int stacksize

} DuStack; （2分）

栈1空条件：s.top[0]=s.base[0] （1分）

栈2空条件：s.top[1]=s.base[1] （1分）

栈满条件：s.top[0]-s.top[1]=1 （1分）

（2）

status gettop(DuStack S, int i, SelemType &e) {

if (i!=0 && i!=1 ) return ERROR; (2分)

switch i {

case 0: e=\*s.top[0];

case 1: e=\*s.top[1];

}

return OK; （3分）

}

3、Error\_code binary\_search\_2(const Ordered\_list &the\_list,const Key &target, int &position)

{

low=1;high=ST.length;

while (low<=high) {

mid=(low + high) /2;

if (key=ST.elem[mid].key) return mid;

else if (key< ST.elem[mid].key) high = mid-1;

else low = mid+1;

}

return 0;

}