

1、(1) 证明: $O(f)+O(g)=O(f+g)$ (7 分)

(2) 求下列函数的渐近表达式: (6 分)

① $3n^2+10n$;

② $21+1/n$;

2、对于下列各组函数 $f(n)$ 和 $g(n)$, 确定 $f(n)=O(g(n))$ 或 $f(n)=\Omega(g(n))$ 或 $f(n)=\theta(g(n))$, 并简述理由。(15 分)

(1) $f(n) = \log n^2; g(n) = \log n + 5$;

(2) $f(n) = \log n^2; g(n) = \sqrt{n}$;

(3) $f(n) = n; g(n) = \log^2 n$;

3、试用分治法对数组 $A[n]$ 实现快速排序。(13 分)

4、试用动态规划算法实现最长公共子序列问题。(15 分)

5、试用贪心算法求解汽车加油问题: 已知一辆汽车加满油后可行驶 n 公里, 而旅途中有若干个加油站。试设计一个有效算法, 指出应在哪些加油站停靠加油, 使加油次数最少。(12 分)

6、试用动态规划算法实现下列问题: 设 A 和 B 是两个字符串。我们要用最少的字符操作, 将字符串 A 转换为字符串 B , 这里所说的字符操作包括:

(1) 删除一个字符。

(2) 插入一个字符。

(3) 将一个字符改为另一个字符。

将字符串 A 变换为字符串 B 所用的最少字符操作数称为字符串 A 到 B 的编辑距离, 记为 $d(A,B)$ 。试设计一个有效算法, 对任给的两个字符串 A 和 B , 计算出它们的编辑距离 $d(A,B)$ 。

(16 分)

7、试用回溯法解决下列整数变换问题: 关于整数 i 的变换 f 和 g 定义如下: $f(i) = 3i; g(i) = \lfloor i/2 \rfloor$ 。对于给定的两个整数 n 和 m , 要求用最少的变换 f 和 g 变换次数将 n 变为 m 。(16 分)

1、(1) 证明: $O(f)+O(g)=O(f+g)$ (7 分)

证明: 令 $F(n)=O(f)$, 则存在自然数 n_1 、 c_1 , 使得对任意的自然数 $n \geq n_1$, 有: $F(n) \leq c_1 f(n)$ (2 分)

同理可令 $G(n)=O(g)$, 则存在自然数 n_2 、 c_2 , 使得对任意的自然数 $n \geq n_2$, 有: $G(n) \leq c_2 g(n)$ (3 分)

令 $c_3 = \max\{c_1, c_2\}$, $n_3 = \max\{n_1, n_2\}$, 则对所有的 $n \geq n_3$, 有:

$$F(n) \leq c_1 f(n) \leq c_3 f(n)$$

$$G(n) \leq c_2 g(n) \leq c_3 g(n) \dots\dots\dots (5 \text{ 分})$$

故有:

$$O(f)+O(g)=F(n)+G(n) \leq c_3 f(n)+c_3 g(n)=c_3(f(n)+g(n))$$

因此有:

$$O(f)+O(g)=O(f+g) \dots\dots\dots (7 \text{ 分})$$

(2) 求下列函数的渐近表达式: (6 分)

① $3n^2+10n$;

② $21+1/n$;

① 因为 $\lim_{n \rightarrow \infty} \frac{(3n^2 + 10n) - 3n^2}{3n^2 + 10n} = 0$; 由渐近表达式的定义易知:

$3n^2$ 是 $3n^2+10n$ 的渐近表达式。..... (3 分)

② 因为 $\frac{21 + \frac{1}{n} - 21}{21 + \frac{1}{n}} \rightarrow 0, n \rightarrow \infty$, 由渐近表达式的定义易知:

21 是 $21 + \frac{1}{n}$ 的渐近表达式。..... (6 分)

说明: 函数 $T(n)$ 的渐近表达式 $t(n)$ 定义为:

$$\frac{T(n) - t(n)}{T(n)} \rightarrow 0, n \rightarrow \infty$$

2、解: 经分析结论为:

(1) $\log n^2 = \theta(\log n + 5)$; (5 分)

(2) $\log n^2 = O(\sqrt{n})$; (10 分)

(3) $n = \Omega(\log^2 n)$; (15 分)

3、解: 用分治法求解的算法代码如下:

int partition(float A[], int p, int r)

```
{
    int i=p, j=r+1;
    float x=a[p];
    while(1){
        while(a[++i]<x);
        while(a[--j]>x);
        if(i>=j) break;
        a[i] ↔ a[j]..... (4 分)
```

```
};
a[p]=a[j];
a[j]=x;
return j;..... (7 分)
```

```

}
void Quicksort(float a[],int p,int r)
{
    if(p<r){
        int q=partition(a,p,r);..... (10 分)
        Quicksort(a,p,q-1);
        Quicksort(a,q+1,r);
    }
};
Quicksort(a,0,n-1);..... (13 分)
4、解：用动态规划算法求解的算法代码如下：
int lcs_len(char* a,char* b,int c[][N])
{
    int m=strlen(a),n=strlen(b),i,j;
    for(i=0;i<=m;i++)c[i][0]=0;
    for(j=1;j<=n;j++)c[0][j]=0;..... (4 分)
    for(i=1;i<=m;i++)
        for(j=1;j<=n;j++)
            if(a[i-1]==b[j-1])c[i][j]=c[i-1][j-1]+1;
            else if(c[i-1][j]>=c[i][j-1])
                c[i][j]=c[i-1][j];
            else c[i][j]=c[i][j-1];..... (7 分)
    return c[m][n];..... (8 分)
};
char* build_lcs(char s[],char* a,char* b)
{
    int k,i=strlen(a),j=strlen(b),c[N][N];
    k=lcs_len(a,b,c);
    s[k]='\0';
    while(k>0){
        if(c[i][j]==c[i-1][j])i--;..... (11 分)
        else if(c[i][j]==c[i][j-1])j--;
        else{
            s[--k]=a[i-1];
            i--;j--;
        }
    }
    return s;..... (15 分)
}
5、解： int greedy(vector<int> x,int n)
{
    int sum=0,k=x.size();
    for(int j=0;j<k;j++)
        if(x[j]>n){
            cout<<"Nosolution"<<endl;
            return-1;..... (6 分)
        }
    for(int i=0,s=0;i<k;i++){
        s+=x[i];
        if(s>n){sum++;s=x[i];}..... (9 分)
    }
    return sum;..... (12 分)
}

```

```
}
```

6、解：此题用动态规划算法求解：

```
int dist()
```

```
{
```

```
    int m=a.size();
```

```
    int n=b.size();
```

```
    vector<int> d(n+1,0);
```

```
    for(int i=1;i<=n;i++)d[i]=i;..... (5 分)
```

```
    for(i=1;i<=m;i++){
```

```
        int y=i-1;
```

```
        for(int j=1;j<=n;j++){
```

```
            int x=y;
```

```
            y=d[j];
```

```
            int z=j>1?d[j-1]:i;..... (10 分)
```

```
            int del=a[i-1]==b[j-1]?0:1;
```

```
            d[j]=min(x+del,y+1,z+1);..... (13 分)
```

```
        }
```

```
    }
```

```
    return d[n];..... (16 分)
```

```
}
```

7、解：解答如下：

```
void compute()
```

```
{
```

```
    k=1;
```

```
    while(!search(1,n)){
```

```
        k++;
```

```
        if(k>maxdep)break;
```

```
        init();
```

```
    },..... (6 分)
```

```
    if(found)output();..... (9 分)
```

```
    else cout<<"NoSolution!"<<endl;
```

```
}
```

```
bool search(int dep,int n)
```

```
{
```

```
    if(dep>k)return false;..... (11 分)
```

```
    for(int i=0;i<2;i++){
```

```
        int n1=f(n,i);t[dep]=i;..... (13 分)
```

```
        if(n1==m||search(dep+1,n1)){
```

```
            found=true;
```

```
            out();
```

```
            return true;
```

```
        }
```

```
    }
```

```
    return false;..... (16 分)
```

```
}
```