

题号	一	二	三	四	五	六	七	八	总分
得分									

考生班级		学 号		姓 名	
------	--	-----	--	-----	--

一、选择题（每题 3 分，共 18 分）

- 下面函数中渐进时间最小的是 D。

A. $T_1(n)=n+n\log n$
B. $T_2(n)=2n+n\log n$

C. $T_3(n)=n^2-\log n$
D. $T_4(n)=n+100\log n$
- 采用动态规划策略求解问题的显著特征是满足**最优子结构性**质，其含义是 B。

A. 当前所做出的决策不会影响后面的决策
B. 原问题的最优解包含其子问题的最优解
C. 问题可以找到最优解，但利用贪心法不能找到最优解
D. 每次决策必须是当前看来最优的决策才可以找到最优解
- 在下列算法设计方法中，B在求解问题的过程中并不从整体最优上加以考虑，而是做出在当前看来是最好的选择。B不能保证求得 0/1 背包问题的最优解。

A. 分治法
B. 贪心法
C. 动态规划法
D. 回溯法

题号	一	二	三	四	五	六	七	八	总分
得分									

考生班级		学 号		姓 名	
------	--	-----	--	-----	--

一、选择题（每题 3 分，共 18 分）

- 下面函数中渐进时间最小的是 D。

A. $T_1(n)=n+n\log n$
B. $T_2(n)=2n+n\log n$

C. $T_3(n)=n^2-\log n$
D. $T_4(n)=n+100\log n$
- 采用动态规划策略求解问题的显著特征是满足**最优子结构性**质，其含义是 B。

A. 当前所做出的决策不会影响后面的决策
B. 原问题的最优解包含其子问题的最优解
C. 问题可以找到最优解，但利用贪心法不能找到最优解
D. 每次决策必须是当前看来最优的决策才可以找到最优解
- 在下列算法设计方法中，B在求解问题的过程中并不从整体最优上加以考虑，而是做出在当前看来是最好的选择。B不能保证求得 0/1 背包问题的最优解。

A. 分治法
B. 贪心法
C. 动态规划法
D. 回溯法

4. 以下算法的时间复杂度是__C__

```
void fun(int n)
{
    for(i=0; i<n; i++)
    {
        for(j=i; j<n; j++)
        {
            S++;
        }
    }
}
```

A. 3 B. n C. n^2 D. $n^2 / 2$

5. __B__算法策略与递归技术的联系最弱。

A. 动态规划 B. 贪心 C. 回溯 D. 分治

二、简答题（每题 10 分，共 30 分）

1. 简述动态规划算法的基本思想及其基本要素。

基本思想：动态规划算法的基本思想与分治法类似，也是将待求解的问题分解为若干个子问题（阶段），按顺序求解子阶段，前一子问题的解，为后一子问题的求解提供了有用的信息。在求解任一子问题时，列出各种可能的局部解，通过决策保留那些有可能达到最优的局部解，丢弃其他局部解。依次解决各子问题，最后一个子问题就是初始问题的解。

基本要素：最优子结构性质和子问题重叠性质是该问题可用动态规划算法求解的基本要素。

1.最优子结构

当问题的最优解包含了其子问题的最优解时，称该问题具有最优子结构性质。问题的最优子结构性质提供了该问题可用动态规划算法求解的重要线索。

在动态规划算法中，利用问题的最优子结构性质，以自底向上的方式递归地从子问题的最优解逐步构造出整个问题的最优解。

2.重叠子问题

可用动态规划算法求解的问题应具备的另一个基本要素是子问题的重叠性质。在用递归算法自顶向下求解问题时，每次产生的子问题并不总是新问题，有些子问题被反复计算多次。动态规划算法正是利用了这种子问题的重叠性质，对每一个子问题只解一次，而将其解保存在一个表格中，当再次需要此子问题时，只要简单地用常数时间查看一下结果。

通常，不同的子问题个数随问题的大小呈多项式增长。因此，用动态规划算法通常只需要多项式时间，从而获得较高的解题效率。

最优化原理：如果问题的最优解所包含的子问题的解也是最优的，就称该问题具有最优子结构，即满足最优化原理。

无后效性：某阶段的状态一旦确定，则此后过程的演变不再受此前各种状态及决策的影响。

有重叠子问题：即子问题之间是不独立的，一个子问题在下一阶段决策中可能被多次使用到。

2. 请阐述分支限界搜索（广度优先搜索）算法的一般模式。

```
bool bfs ()
{
    当队列非空时
    {
        从队列中出队一个节点 x（获取并删除）；           // step1
        for(x 的 所有 可 扩 展 节 点 y)
        // step2
        {
            if（检查可达性通过（包括判重）OK）           //step3           {
                将新扩展节点排入队列                           //step4
                记录 y 的父节点为 x；                           记录到达 y 的步
                数                                           }           // 找到目标，结束搜索
            if（新扩展节点 y == 目标节点）return true; //step5 }           } }
```

3. 写出回溯算法搜索子集树的一般模式。

```
void BackTrack(int t)
{
    if(t>n)
        update(x);
    else
        for(int i=0;i<=1;i++)
        {
            x[t]=i;
            if(constraint(t)&&bound(t))
                BackTrack(t+1);
        }
}
```

三、设计一个算法求解下面的问题，要求先说明用什么算法实现，再写出算法的关键部分，并且要求把相关的重要变量定义为全局变量，并且要求在所有变量定义、重要程序段和子函数（不要求实现）都有详细的说明。（每小题 13 分共 52 分）

1. 素数环问题

1 到 n 这 n 个自然数摆成一个环，使得任意两个相邻的数之和都是素数，有多少种方案。

```
Memset(isprime,0,sizeof(isprime));
Isprime[2]==1;
For(int i=3;i<=10000000;i++)
{
    For(int j=2;j<I;j++)
    {
        If (i%j==0)
        {
            Break;
        }
    }
    If(j==i)
    {
        Isprime[i]==1;
    }
}
A[1]=1;
Usd[1]=1;
Int num=0;
Int dfs(int step)
{
    Int I;
    If(step==n+1&&isprime[a[1]+a[n]])
    {
        Num++;
    }
    Else
    {
        For(i=2;i<=n;i++)
        {
            If(!used[i]&&isprime[i+a[step-1]])
            {
```

```
        Used[i]=1;
        A[step]=I;
        Dfs(step+1);
        Used[i]=0;
    }
}
Return;
}
Cout<<num<<endl;
```

2. 用贪心算法解决活动安排问题：设有待安排的 10 项活动，都要使用某一公共资源，每项活动的开始时间和结束时间如下表所示：

I	1	2	3	4	5	6	7	8	9	10
开始时间	3	0	9	1	10	11	2	5	3	4
结束时间	8	6	13	4	13	12	3	10	5	9

用贪心算法求出最多能安排几项活动。并给模拟程序的运行过程给出答案。

3. 给定数字三角形，找出从第一层到最后一层的一条路，使得路径上的权值之和最小。

输入的数字三角形样例如下：

```

    1
   2 3
  4 5 6
 7 8 9 10
```

动态规划问题

```
#include<iostream>
#include<algorithm>
using namespace std;
int t,n,i,j;
int s[100][100];
int main()
{
    cin>>t;
    while(t-->0)
    {
        cin>>n;
        for(i=1;i<=n;i++)
            for(j=1;j<=n;j++)
            {
                s[i][j]=0;
            }
        for(i=1;i<=n;i++)
        {
            for(j=1;j<=i;j++)
                cin>>s[i][j];
        }
        for(i=n;i>=2;i--)
            for(j=1;j<i;j++)
            {
```

`s[i-1][j]+=max(s[i][j],s[i][j+1]);//从下往上找`

`}`

`cout<<s[1][1]<<endl;`

`}`

`return 0;`

`}`

4. 对 8 个元素的序列 $A = (9, 4, 5, 2, 1, 7, 4, 6)$ 进行从小到大的排序，给出归并排序的基本思想，画出归并排序（递归实现）的求解过程。

答：自顶向下的归并排序过程：

基本思想：利用分支策略

(1) 序列一分为二

(2) 子序列递归排序

(3) 合并有序的子序列

9 4 5 2 1 7 4 6

分解：

9 4 5 2

1 7 4 6

分解：

分解：

9 4

5 2

1 7

4 6

分解：

分解：

分解：

分解：

9 4

5 2

1 7

4 6

合并：

合并

合并

合并

4 9

2 5

1 7

4 6

合并

合并

2 4 5 9

1 4 6 7

合并

1 2 4 4 5 6 7 9