```
1.  int   s[]={0,};//用来表示物品重量,从S[1]存到S[n]
    int   X[]={0} //用来标记放了哪些物品
    int   c ; //背包容量
    int   now=0; //当前背包中的物品重量
    int   nowh=c;当前背包中的物品上界
    void  backtrack(int  i);
          { if(i==n+1)return;
             if(now==c)
            {for(int i=1;i<=n;i++)
               if(X[i]) cout<<i;
               return;}
             if(now+S[i]≤C)
             { now+=S[i];
                X[i]=1;
                backtrack(i+1);
                now-=S[i]
                 X[i]=0;}
              if(nowh-S[i]≥C)
              { nowh-=S[i];
                backtrack(i+1);
                nowh+=S[i];}
              }

        int   main(  )
             { backtrack(1);
               return  0;}
```

```
2、 C[n+1][n+1];// C[i][j]表示第j个人完成第i项工作的费用
    X[n+1] ;//X[i]表示第i项工作由X[i]个人完成.
    best X[]:记录最优X[]
     cost   :记录当前费用
    bestcost=∞:最优(少)费用并初始化为无穷大
    void  swap(int i, int j)
   {int a;
     for(int k=1 ;k≤n;k++)
        { a=C[k][i];
          C[k][i]=C[k][j];
          C[k][j]=a;}
        return; }
     void   backtrack(i)
     { if (i==n+1)
      {if (cost<bestcost)
        {bestcost=cost;
        for(i=1;j<=n;i++)
          bestX[i]=X[i];
         return;}}
       if(cost>bestcost)
           return;
      for(int j=i ;j≤n;j++)
        {cost+=C[i][j]
          swap(i,j);
          X[i]=j;
         backtrack(i+1)
         swap(i,j)
         cost-=C[i][j];}
       }
     int   main( )
     {    backtrack(1);
        for(int i=1;i<=n;i++)
         print("把第%d项工作分给第%d个人做\n",i,X[i]);
         print("需要花费的总费用为%d",bestcost);
```

```
3、 int  d[m][];连接块数组
     int X[n+1]={0,1,…n};//目前电路板的排列
    int bestX[n+1];记录最优排列
    int   nowmax=0;
     int bestmax=∞;
    Void  backtrack(int  i)
  { if(i=n+1)
     {if (nowmax<bestmax)
         for(int j=1;j<=n;i++)
         bestX[i]=X[i];
         bestmax=nowmax;}
     if(nowmax≥bestmax)return;
    for(int j=j;j<=n;j++)
   {  swap(i.j);
      int t=find();
       if(t<nowmax)
        int  m=nowmax;
           nowmax=t;
         backtrack(i+1);
          nowmax=m;
     }
    }
  void swap(int i,int j)
  {  int  t=X[i];
      X[i]=X[j];
      X[j]=t;
      return;}
  Void find( )
  { int  max=0;
    for(int i=1;i<=m;i++)
     { int  low=back(d[i][1])
       int  high=back[d[i][1])
       for(int j=2;d[i][j]!=0;j++)
        {if (back(d[i][j])<low)
              low=back(d[i][j]);
         if(back(d[i][j])>high)
              high=back(d[i][j]);
          max=(high-low>max))?high-low:max;
        }
       return  max;
      }
   void  back(int  t)
   { for(int i=1;i<=n;i++)
       if(X[i]==t)
        return  i;
    }
  int  main()
  { for(int i=1;i<=n;i++)
     printf("第%d个位置放第%d个电路板/n",i,bestX[i]);
    printf("有最小最大长度%d",bestmax);
    return  0;
  }
```