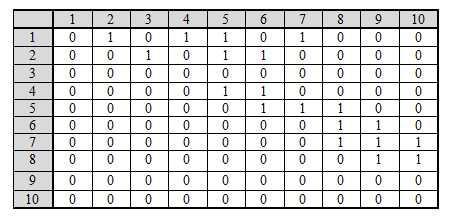
### 魔法石矿★

【算法分析】

可以用一个二维数组来表示矿与矿之间的关系，值为1或0，其中1表示有关联，0表示无关联。如表1.4所示。

表1.4



　　接下来的事情就简单了，模仿求最长不下降序列的方法即可。

　　参考代码如下所示：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50 | //魔法石矿  #include <bits/stdc++.h>  using namespace std;  int a[1001];  int g[1001][1001];  inline bool isNumber(char c)  {  return (c>='0' && c<='9');  }  void GetNum(int n)  {  char c;  for(int i=1; i<=n; i++)  {  int x;  cin>>x;  do  {  int num=0;  while(isNumber(c=getchar()))  num=num\*10+c-'0';  g[x][num]=1;  } while(c!='\n'); //一直读到换行  }  }  int main()  {  int n;  scanf("%d",&n);  for(int i=1; i<=n; ++i)  scanf("%d",&a[i]);  GetNum(n);  int maxn,Ans=0;  for(int i=n; i>=1; i--) //从后往前扫描  {  maxn=0;  for(int j=i+1; j<=n; j++) //找连通路线中的最多魔法石  if(g[i][j] && a[j]>maxn) //连接且值最大  maxn=a[j];  a[i]+=maxn;  Ans=Ans>a[i]?Ans:a[i];  }  printf("%d\n",Ans);  return 0;  } |