算法作业二参考答案

1. 求三个字符串的X、Y、Z的最长公共子序列LCS（X,Y,Z）
2. 写出算法思想与递推式。
3. 写出核心伪代码。
4. 分析算法复杂度。

**参考答案：**

算法思想：

dp[i][j][k]表示A[0...i],B[0...j],C[0...k]的LCS

递推式：

if (a[i]==b[j]&&b[j]==c[k])  
dp[i][j][k]=dp[i-1][j-1][k-1]+1;  
else  
dp[i][j][k]=max(max(dp[i][j][k], dp[i-1][j][k]), max(dp[i][j-1][k], dp[i][j][k-1]));

伪代码如下：

function LCS(A,B,C)

 n1 = A.length;

 n2 =B.length

n3 =C.length

//字符串的个数可以是0-n,共n+1个可能，所以循环中是i<=n

for i=0 to n1

 for j=0 to n2

   for k=0 to n3

     //只要有一个序列的长度为0，公共子序列的长度为0

     if i==0||j==0||k==0 then a[i][j][k] = 0;

 //自底向上求a[i][j][k]

for i=1 to n1

 for j=1 to n2

   for k=1 to n3

      {

         if A[i - 1] == B[j - 1] && B[j - 1] ==C[k - 1])

         then  a[i][j][k] = a[i - 1][j - 1][k - 1] + 1;

        else   a[i][j][k] = max(max(a[i - 1][j][k], a[i][j - 1][k]),a[i][j][k -1]);

      }

return a[n1][n2][n3]

时间复杂度O(n3)

1. 动态数组问题。在数组管理中，我们为一个数组分配一定的内存空间（初始长度为100），随后可能会发现不够用，于是必须为其重新分配更大的空间（之前空间的3倍），并将所有对象从原数组中复制到新的空间中。类似地，如果从数组中删除了很多对象，则为其重新分配一个更小的空间。问题：（1）设计动态数组插入元素的算法伪代码。（2）分析插入*n*个元素的平均时间复杂度（*n*>>100）。
2. 设有*n*个顾客同时等待一项服务。顾客*i*需要的服务时间为*ti*, 1≦*i*≦*n* 。共有*s*处可以提供此服务。应如何安排*n*个顾客的服务次序才能使平均等待时间达到最小?平均等待时间是*n*个顾客等待服务时间的总和除以*n*。

为了平均等待时间最少，当然是用时少的服务先执行，用时长的后执行，排个序就OK了

题**目要求平均等待的时间最短，即时间越短应被计算次数越多，所以对服务时间最短的顾客先服务的贪心选择策略，做完第一次选择后，原问题T变成了需对n—1个顾客服务的新问题T’，规模一直缩小，符合贪心算法。**

double greedy(vector<int>client,int num)//client为顾客等待的duilie，num为窗口总数

{

vector<int>service(num+1,0); //服务窗口每一个顾客等待的时间

vector<int>sum(num+1,0); //服务窗口顾客等待的时间总和

int n=client.size();

sort(client.begin(), client.end());

int i=0,j=0;

while(i<n)

{

service[j]+=client[i];

//很妙的计算，下有附图，可见过程，还有一种思想：sum【j】+=（n/2-i） client【i】，即各个水龙头未接水的后面人都在等，每一次把他们时间都算上

sum[j]+=service[j];

i++;

j++;

if(j==num) j=0; //注意，j==num，不是num-1，因为j++在后

}

double t=0;

for(i=0;i<num;i++)//计算所有窗口时间服务总和

t+=sum[i];

return t/n;

}

int main()

{

int n,i,num;

cin>>n>>num;

vector<int>client(n,0);

for(i=0;i<n;i++)

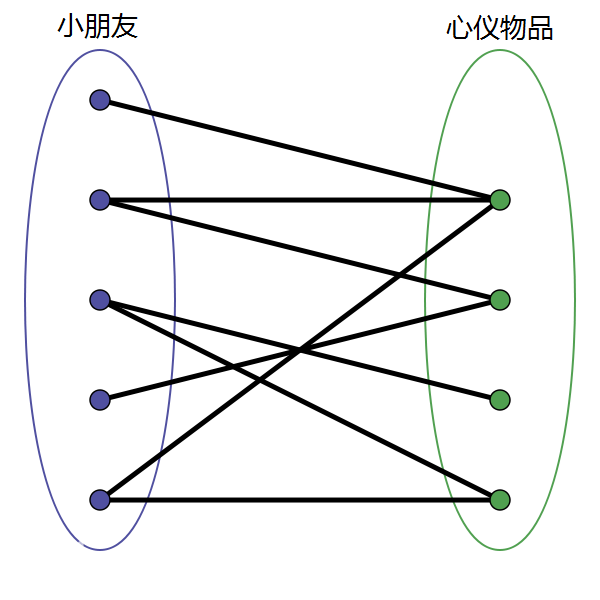
cin>>client[i];

cout<<greedy(client,num);

return 0;

}

1. 现在有一批物资捐赠给孤儿院，捐赠物资数量有限类别较多，每个小朋友均有自己喜欢的若干个物品（关系如下如所示），现在需要设计一个分配策略使得尽量多的小朋友能够分到喜爱的物品（分到一个心仪物品即可）。先写出算法整体思路，然后给出算法伪代码和时间复杂度。



建立源点s到男士集合M（外籍飞行员）的容量为1的有向边（从源点指向M）；

建立女士集合N到汇点t的容量为1的有向边（从N指向汇点）；

根据题中所给映射建立M到N的容量为1的有向边。

求最大流的得到的流量即为最大匹配数。

Ford-Fulkerson (G,s,t)｛

Input:

Output：

1、建立源点s到男士集合M（外籍飞行员）的容量为1的有向边（从源点指向M）；

2、建立女士集合N到汇点t的容量为1的有向边（从N指向汇点）；

3、根据题中所给映射建立M到N的容量为1的有向边。

4、初始化流f为0

While(存在一条增广路径p) {

沿着增广路径p增广流f

}

返回f