**//LCIS**

**/\***

**Parameter: given two vectors**

**Return: longest common increasing subsequence path vector**

**Complexity: O(n\*m)**

**\*/**

vector<int> LCIS\_path(vector<int>v1, vector<int>v2)

{

int n=v1.size(), m=v2.size();

int cur, last, C[m+5], prev[m+5], length=0, index=-1;

vector<int>res;

memset(C, 0, sizeof(C));

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

{

cur=0, last=-1;

for(int j=0;j<m;j++)

{

if (v1[i]==v2[j] && cur>=C[j])

{

C[j]=cur+1;

prev[j]=last;

}

if (v1[i]>v2[j] && cur<C[j])

{

cur=C[j];

last=j;

}

}

}

for(int i=0;i<m;i++)

if (C[i]>length)

{

length=C[i];

index=i;

}

while (index!=-1)

{

res.push\_back(v2[index]);

index=prev[index];

}

reverse(res.begin(),res.end());

return res;

}