

暨南大学本科实验报告专用纸

课程名称 ACM 程序设计实验 成绩评定
实验项目名称 动态规划 指导教师 张震
实验项目编号 0806017905 实验项目类型 综合性 实验地点 116
学生姓名 阮炜霖 学号 2020101603
学院 信息科学技术学院 系 计算机科学系 专业 网络工程
实验时间 2022 年 6 月 1 日 下午~6 月 1 日 下午 温度 °C 湿度

（一）实验目的

- 1) 掌握动态规划算法。
- 2) 能够利用动态规划算法解决实际问题。

（二）实验内容和要求

实验内容::

- 1.HDU 2084
- 2.HDU 1159
- 3.AtCoder-dp_f
- 4.POJ 1651

实验要求:

完成所有题目的在线测评。

（三）主要仪器设备

仪器: 计算机

实验环境: Windows 2000 或 Windows XP + Visual C++6.0 或 Turbo C 2.0

（四）源程序

HDU 2084 数塔

```

1. #include<bits/stdc++.h>
2. #define int long long
3. using namespace std;
4. const int N=107;
5.
6. int n,ans,a[N][N],dp[N][N];
7.
8. void Solve(){
9.     cin>>n;
10.    ans=0;
11.    for(int i=1;i<=n;i++){
12.        for(int j=1;j<=i;j++){
13.            cin>>a[i][j];
14.        }
15.    }
16.    memset(dp,0,sizeof(dp));
17.    dp[1][1]=a[1][1];
18.    for(int i=1;i<n;i++){
19.        for(int j=1;j<=i;j++){
20.            dp[i+1][j]=max(dp[i+1][j],dp[i][j]+a[i+1][j]);
21.            dp[i+1][j+1]=max(dp[i+1][j+1],dp[i][j]+a[i+1][j+1]);
22.        }
23.    }
24.    for(int i=1;i<=n;i++) ans=max(ans,dp[n][i]);
25.    cout<<ans<<"\n";
26. }
27.
28. signed main(){
29. // ios::sync_with_stdio(0);
30. // cin.tie(0);cout.tie(0);
31. int T=1;
32. cin>>T;
33. while(T--){
34.     Solve();
35. }
36. return 0;
37. }

```

HDU 1159 Common Subsequence

```

1. // #pragma GCC optimize("Ofast", "inline", "-f
    fast-math")

```

```

2.          // #pragma GCC target("avx,sse2,sse3,sse4,mmx"
3.          )
4.          #include<cstring>
5.          #include<algorithm>
6.          #include<iostream>
7.          using namespace std;
8.          const int mod=1e9+7;
9.          //int read(){  int x=0,f=1;char ch=getchar()
10.         ;while(ch<'0' || ch>'9'){if(ch=='-') f=f*-1;ch=getchar();}while(ch>='0'
11.         &&ch<='9'){x=x*10+ch-'0';ch=getchar();}return x*f;}
12.         //void write(int x){if(x>9) write(x/10);putch
13.         ar(x%10+'0');}
14.         string S,T;
15.         int n,m,dp[1005][1005];
16.         signed main(){
17.         // ios::sync_with_stdio(0);
18.         // cin.tie(0);cout.tie(0);
19.         // freopen("in.cpp","r",stdin);
20.         // freopen("out.cpp","w",stdout);
21.         // cin>>T;
22.         // clock_t start,finish;
23.         // start=clock();
24.         while(cin>>S>>T){
25.             memset(dp,0,sizeof(dp));
26.             n=S.length(),m=T.length();
27.             for(int i=1;i<=n;++i){
28.                 for(int j=1;j<=m;++j){
29.                     dp[i][j]=max(dp[i-1][j],dp[i]
30.                     [j-1]);
31.                     if(S[i-1]==T[j-1]){
32.                         dp[i][j]=max(dp[i][j],dp[
33.                         i-1][j-1]+1);
34.                     }
35.                 }
36.             }
37.             cout<<dp[n][m]<<"\n";
38.         }
39.         // finish=clock();
40.         // cerr<<((double)finish-start)/CLOCKS_PER_S
41.         EC<<endl; return 0;
42.     }

```

AtCoder-dp_f LCS

```
1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<cstring>
4. #include<algorithm>
5. #include<iostream>
6. #define inf 0x3f3f3f3f
7. #define int long long
8. using namespace std;
9. const int N=2e5+7;
10. const int mod=1e9+7;
11.
12. //int read(){ int x=0,f=1;char ch=getchar();while(ch<'0' || ch>'9'){i
    f(ch=='-') f=f*-1;ch=getchar();}while(ch>='0'&&ch<='9'){x=x*10+ch-'0'
    ;ch=getchar();}return x*f;}
13. //void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
14.
15. string S,T;
16. char stk[3005];
17. int n,m,top=0,dp[3005][3005];
18. pair<int,int>pre[3005][3005];
19.
20. void Solve(){
21.     cin>>S>>T;
22.     n=S.length(),m=T.length();
23.     for(int i=1;i<=n;++i){
24.         for(int j=1;j<=m;++j){
25.             if(dp[i-1][j]>dp[i][j-1]) dp[i][j]=dp[i-1][j],pre[i][j]=m
                ake_pair(i-1,j);
26.             else dp[i][j]=dp[i][j-1],pre[i][j]=make_pair(i,j-1);
27.             if(S[i-1]==T[j-1]){
28.                 if(dp[i-1][j-1]+1>dp[i][j]){
29.                     dp[i][j]=dp[i-1][j-1]+1;
30.                     pre[i][j]=make_pair(i-1,j-1);
31.                 }
32.             }
33.         }
34.     }
35.     int p1=n,p2=m;
36.     while(1){
37.         if(!p1&&!p2) break;
38.         int x=pre[p1][p2].first,y=pre[p1][p2].second;
39.         p1=x,p2=y;
40.         if(S[x]==T[y]) stk[++top]=S[x];
```

```

41.     }
42.     for(int i=top;i>=1;--i) cout<<stk[i];
43.     //cout<<dp[n][m]<<"\n";
44. }
45.
46. signed main(){
47. // ios::sync_with_stdio(0);
48. // cin.tie(0);cout.tie(0);
49. // freopen("in.cpp","r",stdin);
50. // freopen("out.cpp","w",stdout);
51.     int T=1;
52. // cin>>T;
53. // clock_t start,finish;
54. // start=clock();
55.     while(T--){
56.         Solve();
57.     }
58. // finish=clock();
59. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
60. }

```

POJ 1651 Multiplication Puzzle

```

1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<cstring>
4. #include<algorithm>
5. #include<iostream>
6. using namespace std;
7. const int mod=1e9+7;
8.
9. //int read(){ int x=0,f=1;char ch=getchar();while(ch<'0' || ch>'9'){if(
    ch=='-') f=f*-1;ch=getchar();}while(ch>='0'&&ch<='9'){x=x*10+ch-'0';c
    h=getchar();}return x*f;}
10. //void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
11.
12. string S,T;
13. int n,m,dp[1005][1005];
14. signed main(){
15. // ios::sync_with_stdio(0);
16. // cin.tie(0);cout.tie(0);
17. // freopen("in.cpp","r",stdin);
18. // freopen("out.cpp","w",stdout);
19. // cin>>T;

```

```

20. // clock_t start,finish;
21. // start=clock();
22. while(cin>>S>>T){
23.     memset(dp,0,sizeof(dp));
24.     n=S.length(),m=T.length();
25.     for(int i=1;i<=n;++i){
26.         for(int j=1;j<=m;++j){
27.             dp[i][j]=max(dp[i-1][j],dp[i][j-1]);
28.             if(S[i-1]==T[j-1]){
29.                 dp[i][j]=max(dp[i][j],dp[i-1][j-1]+1);
30.             }
31.         }
32.     }
33.     cout<<dp[n][m]<<"\n";
34. }
35. // finish=clock();
36. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
37. }

```

（五）实验步骤与调试

HDU 2084 数塔

找一条路径使得路上的数字加起来最大。可知数塔上每个点都只由前面两个数字转移过来，直接前缀和转移然后取最大值即可。

HDU 1159 Common Subsequence

找两个字符串的最长子序列，直接 $O(n*m)$ 去转移， $dp[i][j]$ 表示 A 串前 i 个位置，B 串前 j 个的最长子序列长度，那么直接由 $dp[i-1][j-1]$, $dp[i-1][j]$, $dp[i][j-1]$ 三个状态转移得来。

AtCoder-dp_f LCS

第二个程序的改版，因为要输出方案，所以把每个状态的前驱记一下即可。注意记录当前状态转移有无新添字符。

POJ 1651 Multiplication Puzzle

每次操作抽取序列中间的一个数 x 的代价为 $a[x]*a[x-1]*a[x+1]$ ，题目问抽到剩下首尾两个数时的最小代价。直接区间 DP 去做就好了， $dp[s][e]$ 表示区间 s 到 e 合并剩首尾时的最小

代价，那么转移方程便是：

```
dp[st][ed]=min(dp[st][ed],dp[st][k]+dp[k][ed]+a[k]*a[st]*a[ed]);
```

（六）实验结果与分析

HDU 2084 数塔

做过无数遍的题了，水过。

[#36677712](#) | [LINNO's solution for \[HDU-2084\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	187ms	1588kB	620	G++	2022-06-15 01:33:17	<input checked="" type="checkbox"/>

HDU 1159 Common Subsequence

讲过的题，水过。

[#36677842](#) | [LINNO's solution for \[HDU-1159\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	109ms	5344kB	1020	G++	2022-06-15 02:02:00	<input checked="" type="checkbox"/>

AtCoder-dp_f LCS

输出方案比较麻烦，直接开 pair 记录每个状态的前驱即可。

[#36677795](#) | [LINNO's solution for \[AtCoder-dp_f\]](#)

Status	Time	Memory	Length	Lang	Submitted
Accepted	159ms	214544kB	1485	C++ (Clang 10.0.0)	2022-06-15 01:51:51

POJ 1651 Multiplication Puzzle

区间 DP，经典套路：枚举长度->枚举起点->枚举中间点->更新区间答案，水过。

[#36679439](#) | [LINNO's solution for \[POJ-1651\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	32ms	1040kB	1308	G++	2022-06-15 10:06:19	<input checked="" type="checkbox"/>