//dp最长不降子序列

rep(i, 1, n)

{dp[i][0] = dp[i][1] = 1;

rep(j, 1, i-1)

{

if(a[j] > a[i]) dp[i][0] = max(dp[i][0], dp[j][1]+1);

if(a[j] < a[i]) dp[i][1] = max(dp[i][1], dp[j][0]+1);

}

maxn = max(maxn, max(dp[i][0], dp[i][1]));}

cout << maxn << endl;

/\*

设dp[i]表示前i个字符能凑成上升子序列的最大长度

dp[i] = max(dp[j]+1, dp[i])

【j < i && s[j] < s[i]】

\*/

for(int i = 1; i <= n; i++) dp[i] = 1;

for(int i = 1; i <= n; i++)

{for(int j = 1; j < i; j++)

if(a[j] < a[i]) dp[i] = max(dp[i], dp[j] + 1);

ans = max(ans, dp[i]);

}