

Tyvj p1048 田忌赛马

解题报告：

首先看到这道题，大部分人应该还是没有思路的，但熟知历史故事的人可能会想到贪心策略。贪心规则如下：将最双方的马排序，如果自己最强的马还拼不过对方，就用最弱的马输掉。这个贪心策略乍一看上去是对的，但经过自己的数据可以看出，这个贪心策略没有考虑到如果存在打平该怎么办。一旦出现可以打平的情况，则会导致两种分支，要不是打平，要不然为了以后得分更高此时输掉。。

不过，这个贪心策略倒是提醒我们可以使用动态规划进行求解，虽然齐王进行比赛出马的实力是不确定的，但是我们仍然可以通过排序使其有序化，强制齐王按照马的实力从大到小进行出马。而根据贪心，我们每次可以从同样经过排序后的田忌的赛马顺序的最左端或最右端出马，每次出完马后算出所有的情况，供后面调用（满足最有子结构性质且无后效性）

则方程如下：

$$F[i, j] = \begin{cases} F[i-1, 0] + \text{compare}(n-i, i) & (j == 0) \\ \max(F[i-1, j-1] + \text{compare}(j, i), F[i-1, j] + \text{compare}(n-i+j, i)) & (0 < j < i) \\ F[i-1, i-1] + \text{compare}(i, i) & (j == i) \end{cases}$$

$F[i, j]$ 表示在第 i 轮比赛中田忌已经从最右端（最大的马的地方）选取了 j 匹马，从最左端选取了 $n-i+j$ 匹马时的最大奖金。

其中，当 $j=0||j=i$ 时需要特判（贡献 1wa）

代码如下：

```
#include "iostream"
```

```
#include "cstring"
```

```
#include "cstdio"
```

```
#include "cstdlib"
```

```

#include "algorithm"

using namespace std;

int n;

int tianji[1001];

int qiwang[1001];

int v[1001][1001];

inline int qusi(int i,int j)
{
    if (v[i][j]!=-1) return v[i][j];

    int ret=tianji[i]>qiwang[j]?200:-200;

    if (tianji[i]==qiwang[j]) ret=0;

    return v[i][j]=ret;
}

inline bool cmp(int a,int b) {return a>b;;}

int F[1001][1001];

int main()
{
    scanf("%d",&n);

    for (int i=1;i<=n;i++) scanf("%d",&tianji[i]);

    for (int i=1;i<=n;i++) scanf("%d",&qiwang[i]);

    sort(tianji+1,tianji+1+n,cmp);sort(qiwang+1,qiwang+1+n,cmp);

    memset(v,-1,sizeof v);

```

```

for (int i=1;i<=n;i++)
{
    F[i][i]=F[i-1][i-1]+qusi(i,i);

    for (int j=i-1;j>=1;j--)

        F[i][j]=max(F[i-1][j-1]+qusi(j,i),F[i-1][j]+qusi(n-i+j,i));

    F[i][0]=F[i-1][0]+qusi(n-i,i);
}

int mx=-1000000000;

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

    mx=max(mx,F[n][i]);

cout<<mx<<endl;

return 0;

}

```