

# 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 ACM 入门 指导教师 张震  
实验项目编号 0806017901 实验项目类型 验证性 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) 熟悉 HDU OJ 的调试步骤和方法;
- 2) 掌握 HDU OJ 的输入输出方法。

## (二) 实验内容和要求

实验内容:

1. HDU 1061
2. HDU 1096

实验要求:

1. 完成所有题目的在线测评。调试通过所有题目。
2. 了解 OJ 中出现的常见错误, 并懂得如何解决。

## (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-1061 Rightmost Digit

```
1. //#pragma GCC optimize("Ofast", "inline", "-ffast-math")  
2. //#pragma GCC target("avx,sse2,sse3,sse4,mmx")
```

```

3. #include <bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. #define lb (x&(-x))
7. #define ls (p<<1)
8. #define rs (p<<1|1)
9. using namespace std;
10. const int N=2e5+7;
11. const int mod=1e9+7;
12.
13. //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;}
14. //void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
15.
16. int mp[20],lst[20];
17.
18. void Solve(){
19.     int idx=0,n,tmp=1,r,ans=1;
20.     memset(mp,0,sizeof(mp));
21.     memset(lst,0,sizeof(lst));
22.     cin>>n;
23.     while(1){
24.         tmp=tmp*n%10;
25.         mp[++idx]=tmp;
26.         if(lst[tmp]){
27.             r=idx-lst[tmp]; //一个周期的长度
28.             r=(n-1)%r+1; //剩余多少次变化
29.             break;
30.         }else lst[tmp]=idx;
31.     }
32.     for(int i=1;i<=r;++i){
33.         ans=ans*n%10;
34.     }
35.     cout<<ans<<"\n";
36. }
37.
38. signed main(){
39.     // ios::sync_with_stdio(0);
40.     // cin.tie(0);cout.tie(0);
41.     // freopen("in.cpp","r",stdin);
42.     // freopen("out.cpp","w",stdout);
43.     int T=1;
44.     cin>>T;

```

```

45. // clock_t start,finish;
46. // start=clock();
47. while(T--){
48.     Solve();
49. }
50. // finish=clock();
51. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
52. }

```

#### HDU-1096 A+B for Input-Output Practice (VIII)

```

1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. //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;}
11. //void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
12. int n,m;
13.
14. void Solve(){
15.     cin>>n;
16.     for(int i=1;i<=n;++i){
17.         cin>>m;
18.         int sum=0;
19.         for(int j=1,x;j<=m;++j){
20.             cin>>x;
21.             sum+=x;
22.         }
23.         cout<<sum<<"\n";
24.         if(i!=n) cout<<"\n";
25.     }
26. }
27.
28. signed main(){
29. // ios::sync_with_stdio(0);
30. // cin.tie(0);cout.tie(0);
31. // freopen("in.cpp","r",stdin);
32. // freopen("out.cpp","w",stdout);

```

```

33.     int T=1;
34. //   cin>>T;
35. //   clock_t start,finish;
36. //   start=clock();
37.     while(T--){
38.         Solve();
39.     }
40. //   finish=clock();
41. //   cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl;   return 0;
42. }

```

## （五）实验步骤与调试

### HDU-1061 Rightmost Digit

题目要求给定  $n$  时， $n^n$  的最后一个数位是多少。我们知道乘积最后一位结果只跟乘数和被乘数的最后一位有关（列式子易得），那么我们可以推得， $A=B \times N \% 10$  表示第  $i$  项的最后一位为  $A$  的结果是第  $i-1$  项最后一位为  $B$  乘以  $N$  的最后一位，显然 10 步以内必有周期，我们忽略计算重复的周期就可以避免超时得到答案。

### HDU-1096 A+B for Input-Output Practice (VIII)

题目给定  $n$  组数据，每组数据给定  $m$  以及接下来的  $m$  个数字，对每组数字求和。开个 `long long` 就可以解决越界问题。注意最后不能输出多余空行。

## （六）实验结果与分析

### HDU-1061 Rightmost Digit

使用一个数记录当前最后一位是多少，然后反复乘  $n$ ，直到重复为止，求得周期。取这个周期多的一段来乘，作为答案即可。做法二：这道题好像直接一个快速幂就完事了。

Username	OJ	Prob	Result	Submit Time
<input type="text"/>	HDU	1061	All	
LINNO	HDU	1061	Accepted	2022-06-14 14:58:34

### HDU-1096 A+B for Input-Output Practice (VIII)

非常简单的模拟。

Username	OJ	Prob	Result	Submit Time
<input type="text"/>	HDU	1096	All	
LINNO	HDU	1096	Accepted	2022-06-14 15:06:58

# 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 数学问题 指导教师 张震  
实验项目编号 0806017902 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) 了解基本数学问题的求解方法;
- 2) 掌握基本数学方法解决实际问题。

## (二) 实验内容和要求

实验内容::

- 1.HDU 1021
- 2.HDU 1060
- 3.CodeForces-1625B

实验要求:

1. 调试通过所有题目。
2. 了解 OJ 中出现的常见错误, 并懂得如何解决。

## (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-1021 Fibonacci Again

```
1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define int long long
5. using namespace std;
6. const int N=2e6+7;
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. int n,fib[N];
12.
13. signed main(){
14. // ios::sync_with_stdio(0);
15. // cin.tie(0);cout.tie(0);
16. // freopen("in.cpp","r",stdin);
17. // freopen("out.cpp","w",stdout);
18. int T=1;
19. // cin>>T;
20. // clock_t start,finish;
21. // start=clock();
22. fib[0]=7%3;fib[1]=11%3;
23. for(int i=2;i<=1000000;++i){
24.     fib[i]=fib[i-1]+fib[i-2];
25.     fib[i]%=3;
26. }
27. while(cin>>n){
28.     if(fib[n]) puts("no");
29.     else puts("yes");
30. }
31. // finish=clock();
32. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
33. }
```

## 1.HDU-1060 Leftmost Digit

```
    //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. //define inf 0x3f3f3f3f
5. #define int long long
```

```

6. // #define int __int128
7. using namespace std;
8. const int N=2e5+7;
9. const int mod=1e9+7;
10.
11. 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';ch=
    getchar();}return x*f;}
12. void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
13.
14. inline int fpow(int a,int b){
15.     int res=1;
16.     while(b){
17.         if(b&1) res=res*a;
18.         a=a*a;
19.         while(res>1000000000) res/=10;
20.         while(a>1000000000) a/=10;
21.         b>>=1;
22.     }
23.     return res;
24. }
25.
26. void Solve(){
27.     double n,m;
28.     cin>>n;
29.     m=n*log10(n);
30.     m-=(int)m;
31.     cout<<(int)pow(10,m)<<"\n";
32. }
33.
34. signed main(){
35.     // ios::sync_with_stdio(0);
36.     // cin.tie(0);cout.tie(0);
37.     // freopen("in.cpp","r",stdin);
38.     // freopen("out.cpp","w",stdout);
39.     int T=1;
40.     T=read();
41.     // clock_t start,finish;
42.     // start=clock();
43.     while(T--){
44.         Solve();
45.     }
46.     // finish=clock();
47.     // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;

```

```
48. }
```

#### CodeForces-1625B Elementary Particles

```
1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include <bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. // 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';ch=getchar();}return x*f;}
11. int t,n,ans;
12.
13. signed main(){
14. // ios::sync_with_stdio(0);
15. // cin.tie(0);cout.tie(0);
16. // freopen("in.cpp","r",stdin);
17. // freopen("out.cpp","w",stdout);
18. cin>>t;
19. while(t--){
20. cin>>n;
21. ans=-1;
22. vector<pair<int,int> >v;
23. for(int i=0;i<n;i++){
24. int num;
25. cin>>num;
26. v.push_back(make_pair(num,i));
27. }
28. sort(v.begin(),v.end());
29. for(int i=0;i<n-1;i++){
30. if(v[i].first==v[i+1].first){
31. ans=max(ans,n-v[i+1].second+v[i].second);
32. }
33. }
34. cout<<ans<<"\n";
35. }
36. return 0;
37. }
```



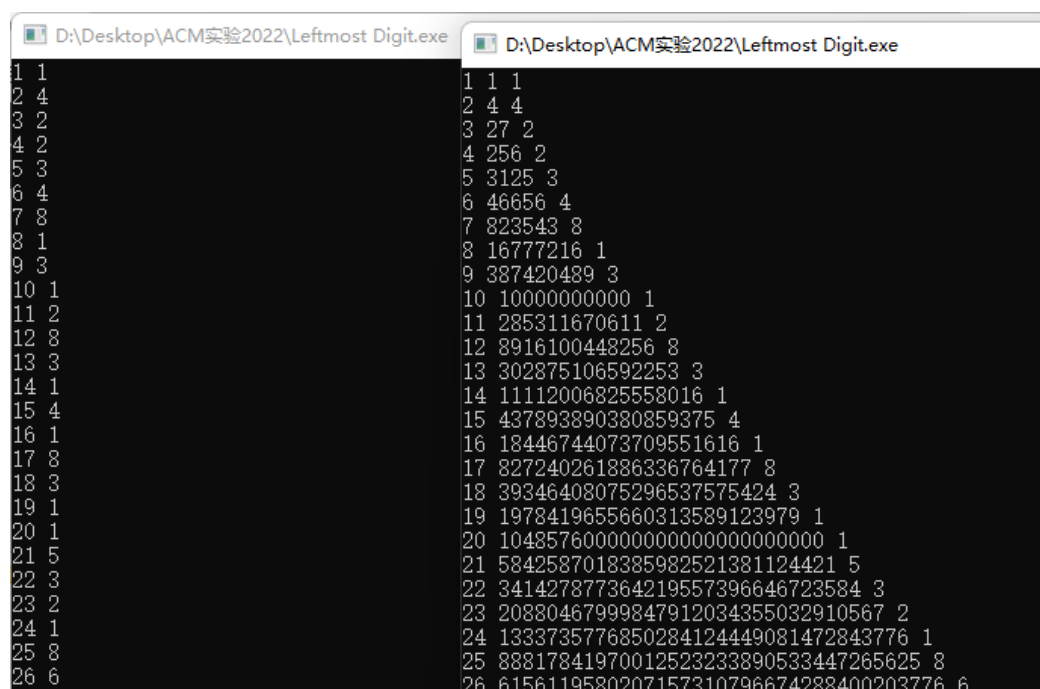
## （五）实验步骤与调试

### HDU-1021 Fibonacci Again

题目给定一个新的斐波那契递推式，问我们第  $n$  项能否被 3 整除。我们看一眼数据范围，1000000 项的斐波那契是会爆 long long 的，不可以直接判断。我们根据取模运算的运算律可以边推边取模，这样就可以避免这个问题。预处理好  $1 \sim 1e6$  以内的所有项，就可以直接输出询问了。

### HDU-1060 Leftmost Digit

题目要我们求  $N^n$  的最左边一个数位。直接暴力连  $n=30$  都过不了，但是可以打表验证。感觉最左边的结果只会与乘数和被乘数最左边的 9 位数有关，所以修改了一下快速幂，保留最右边的 9 位数，然后继续相乘。通过 int128 打了个表，验证了一下应该是对的。



N	$N^N$	Leftmost Digit
1	1	1
2	4	4
3	27	2
4	256	2
5	3125	3
6	46656	4
7	823543	8
8	16777216	1
9	387420489	3
10	1000000000	1
11	285311670611	2
12	8916100448256	8
13	302875106592253	3
14	11112006825558016	1
15	437893890380859375	4
16	18446744073709551616	1
17	827240261886336764177	8
18	39346408075296537575424	3
19	1978419655660313589123979	1
20	1048576000000000000000000	1
21	5842587018385982521381124421	5
22	341427877364219557396646723584	3
23	20880467999847912034355032910567	2
24	1333735776850284124449081472843776	1
25	88817841970012523233890533447265625	8
26	6156119580207157310796674288400203776	6

交上去发现 WA 了，于是赶紧改成了对数形式，通过将  $\text{ans} = n^n$  转化成  $\log_{10}(\text{ans}) = n * \log_{10}(n)$ ，成功将答案的 10 的幂次数搞了出来，那其实对次数做浮点运算之后得到最高位是 10 的多少次幂，就能得到最高位的结果了。

### CodeForces-1625B Elementary Particles

给定一个序列，如果找到一对子区间，在相同的位置上由相同的数，则这一对是合法的。问最长的合法序列对的长度。显然我们只需要关注每一对相同的数字，相同的两个数字向左右延伸最长的相同长度就是我们要的答案，而这个答案显然只于到达整个序列的两端的步数有关。直接做，答案取  $n - \text{pos1} + \text{pos2}$  的最大值就行了，其中  $\text{pos1}$  为相同数字中靠前的一个位置， $\text{pos2}$  是靠后的位置。

## （六）实验结果与分析

### HDU-1021 Fibonacci Again

在斐波那契递推的基础上，对每一次加法运算之后都取模 3 即可。

[#36673013](#) | [LINNO's solution for \[HDU-1021\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	31ms	9224kB	902	G++	2022-06-14 15:18:32	<input checked="" type="checkbox"/>

### HDU-1060 Leftmost Digit

取对数后次数相减就知道最高数位是 10 的多少次幂了，要注意取整。

[#36673316](#) | [LINNO's solution for \[HDU-1060\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	15ms	1464kB	1065	G++	2022-06-14 16:10:51	<input checked="" type="checkbox"/>

### CodeForces-1625B Elementary Particles

直接对相同的两个数套一下  $\text{ans}=\max\{n-\text{pos1}+\text{pos2}\}$  的公式即可。

[#36673483](#) | [LINNO's solution for \[CodeForces-1625B\]](#)

Status	Time	Memory	Length	Lang
Accepted	405ms	7188kB	883	GNU G++20 11.2.0 (64 bit, winlibs)

# 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 递推求解 指导教师 张震  
实验项目编号 0806017903 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   ℃湿度   

## (一) 实验目的

- 1) 熟悉 HDU OJ 的调试步骤和方法。
- 2) 掌握递推求解算法。

## (二) 实验内容和要求

实验内容::

- 1.HDU 2046
- 2.HDU 2045
- 3.CodeForces-1625B

实验要求:

1. 调试通过所有题目。
2. 了解 OJ 中出现的常见错误, 并懂得如何解决。

## (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-2045 不容易系列之(3)—— LELE 的 RPG 难题

```
1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include <bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. int f[55][5][5],n;
11.
12. signed main(){
13. // ios::sync_with_stdio(0);
14. // cin.tie(0);cout.tie(0);
15. // freopen("in.cpp","r",stdin);
16. // freopen("out.cpp","w",stdout);
17. int T=1;
18. // cin>>T;
19. // clock_t start,finish;
20. // start=clock();
21. f[1][0][0]=f[1][1][1]=f[1][2][2]=1;
22. for(int i=2;i<=55;++i){
23. f[i][0][0]=f[i-1][0][1]+f[i-1][0][2];
24. f[i][0][1]=f[i-1][0][0]+f[i-1][0][2];
25. f[i][0][2]=f[i-1][0][0]+f[i-1][0][1];
26. f[i][1][0]=f[i-1][1][1]+f[i-1][1][2];
27. f[i][1][1]=f[i-1][1][0]+f[i-1][1][2];
28. f[i][1][2]=f[i-1][1][0]+f[i-1][1][1];
29. f[i][2][0]=f[i-1][2][1]+f[i-1][2][2];
30. f[i][2][1]=f[i-1][2][0]+f[i-1][2][2];
31. f[i][2][2]=f[i-1][2][0]+f[i-1][2][1];
32. }
33. while(cin>>n){
34. if(n==0) cout<<1<<"\n";
35. else if(n==1) cout<<3<<"\n";
36. else{
37. int ans=f[n][0][1]+f[n][0][2]+f[n][1][0]+f[n][1][2]+f[n][2][0]+f[n][2][1];
38. cout<<ans<<"\n";
39. }
40. }
41. // finish=clock();
42. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
```

```
43. }
```

#### HDU-2046 骨牌铺方格

```
1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. //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;}
11. //void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
12.
13. int n,fib[105];
14.
15. signed main(){
16. // ios::sync_with_stdio(0);
17. // cin.tie(0);cout.tie(0);
18. // freopen("in.cpp","r",stdin);
19. // freopen("out.cpp","w",stdout);
20. int T=1;
21. // cin>>T;
22. // clock_t start,finish;
23. // start=clock();
24. fib[0]=1;fib[1]=1;fib[2]=2;
25. for(int i=3;i<=55;++i) fib[i]=fib[i-1]+fib[i-2];
26. while(cin>>n){
27. cout<<fib[n]<<"\n";
28. }
29. // finish=clock();
30. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
31. }
```

#### CodeForces-1625B Elementary Particles

```
1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
```

```

7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. //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;}
11. int t,n,ans;
12.
13. signed main(){
14. // ios::sync_with_stdio(0);
15. // cin.tie(0);cout.tie(0);
16. // freopen("in.cpp","r",stdin);
17. // freopen("out.cpp","w",stdout);
18. cin>>t;
19. while(t--){
20.     cin>>n;
21.     ans=-1;
22.     vector<pair<int,int> >v;
23.     for(int i=0;i<n;i++){
24.         int num;
25.         cin>>num;
26.         v.push_back(make_pair(num,i));
27.     }
28.     sort(v.begin(),v.end());
29.     for(int i=0;i<n-1;i++){
30.         if(v[i].first==v[i+1].first){
31.             ans=max(ans,n-v[i+1].second+v[i].second);
32.         }
33.     }
34.     cout<<ans<<"\n";
35. }
36. return 0;
37. }

```

## （五）实验步骤与调试

HDU-2045 不容易系列之(3)—— LELE 的 RPG 难题

有  $n$  个连续的方格，每个格子可以涂 3 种不同的颜色，要求相邻两个格子不同色，首尾也不同色，方案数。特判掉  $n=0$  时有 1 种方案， $n=1$  时有 3 种方案（这个时候首尾不同色这个条件就没用了），然后直接动态规划就行了。 $dp[i][j][k]$  表示第一个格子是  $j$  颜色，第  $i$  个格子是  $k$  颜色的方案数，那么就可以由  $dp[i-1][j]$  中其余两种颜色方案数转移过来。

### HDU-2046 骨牌铺方格

有  $2 \times n$  的格子，给定  $1 \times 2$  的块可以横着放和竖着放，问放法方案数。显然由前面  $2 \times (n-1)$  格子和  $2 \times (n-2)$  格子递推得来，因为对于  $1 \times 2$  只能竖着放 1 格，对于  $2 \times 2$  可以横着或者竖着放 2 个。

### CodeForces-1625B Elementary Particles

给定一个序列，如果找到一对子区间，在相同的位置上由相同的数，则这一对是合法的。问最长的合法序列对的长度。显然我们只需要关注每一对相同的数字，相同的两个数字向左右延伸最长的相同长度就是我们要的答案，而这个答案显然只于到达整个序列的两端的步数有关。直接做，答案取  $n - \text{pos1} + \text{pos2}$  的最大值就行了，其中  $\text{pos1}$  为相同数字中靠前的一个位置， $\text{pos2}$  是靠后的位置。

## （六）实验结果与分析

### HDU-2045 不容易系列之(3)—— LELE 的 RPG 难题

DP 水题，开个三维数组直接做就行了。

[#36673748 | LINNO's solution for \[HDU-2045\]](#)

Status	Memory	Length	Lang	Submitted	Open	Share text ?
Accepted	1408kB	1153	G++	2022-06-14 17:14:37	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### HDU-2046 骨牌铺方格

递推斐波那契，裸题

[#36673671 | LINNO's solution for \[HDU-2046\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	15ms	1400kB	884	G++	2022-06-14 16:57:35	<input checked="" type="checkbox"/>

### CodeForces-1625B Elementary Particles

直接对相同的两个数套一下  $\text{ans} = \max\{n - \text{pos1} + \text{pos2}\}$  的公式即可。

[#36673483 | LINNO's solution for \[CodeForces-1625B\]](#)

Status	Time	Memory	Length	Lang
Accepted	405ms	7188kB	883	GNU G++20 11.2.0 (64 bit, winlibs)

# 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 贪心算法 指导教师 张震  
实验项目编号 0806017904 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) 掌握贪心算法原理。
- 2) 学会如何使用贪心算法解决实际问题。

## (二) 实验内容和要求

实验内容::

- 1.HDU 2037
- 2.HDU 1052
- 3.LibreOJ-2123

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-2037 今年暑假不 AC



```

1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=207;
8. const int mod=1e9+7;
9.
10. //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;}
11. //void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
12. struct node{
13.     int l,r;
14. }s[N];
15.
16. bool cmp(node A,node B){
17.     return A.r<B.r;
18. }
19.
20. int n;
21.
22. void Solve(){
23.     for(int i=1;i<=n;++i){
24.         cin>>s[i].l>>s[i].r;
25.     }
26.     sort(s+1,s+1+n,cmp);
27.     int now=0,ans=0;
28.     for(int i=1;i<=n;++i){
29.         if(s[i].l>=now) ++ans,now=s[i].r;
30.     }
31.     cout<<ans<<"\n";
32. }
33.
34. signed main(){
35. // ios::sync_with_stdio(0);
36. // cin.tie(0);cout.tie(0);
37. // freopen("in.cpp","r",stdin);
38. // freopen("out.cpp","w",stdout);
39.     int T=1;
40. // cin>>T;
41. // clock_t start,finish;
42. // start=clock();

```

```

43.     while(cin>>n){
44.         if(!n) break;
45.         Solve();
46.     }
47. // finish=clock();
48. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
49. }

```

#### HDU-1052 Tian Ji -- The Horse Racing

```

1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. //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;}
11. //void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
12.
13. int n,a[N],b[N];
14.
15. void Solve(){
16.     for(int i=1;i<=n;++i) cin>>a[i];
17.     for(int i=1;i<=n;++i) cin>>b[i];
18.     sort(a+1,a+1+n);
19.     sort(b+1,b+1+n);
20.     int L1=1,L2=1,R1=n,R2=n,ans=0;
21.     for(int i=1;i<=n;++i){
22.         if(a[R1]>b[R2]) --R1,--R2,ans+=200;
23.         else if(a[L1]>b[L2]) ++L1,++L2,ans+=200;
24.         else if(a[L1]<b[R2]) ++L1,--R2,ans-=200;
25.     }
26.     cout<<ans<<"\n";
27. }
28.
29. signed main(){
30. // ios::sync_with_stdio(0);
31. // cin.tie(0);cout.tie(0);
32. // freopen("in.cpp","r",stdin);

```

```

33. // freopen("out.cpp","w",stdout);
34.   int T=1;
35. // cin>>T;
36. // clock_t start,finish;
37. // start=clock();
38.   while(cin>>n){
39.       if(n==0) break;
40.       Solve();
41.   }
42. // finish=clock();
43. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
44. }

```

### LibreOJ-212 最短不公共子串

```

1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. //define int long long
6. using namespace std;
7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. int lena,lenb,lst,sz,root,pre[N],ch[N][26],dp[N<<1],sum[N],mp[N<<1];
11.
12. char A[N],B[N];
13. struct Tree{int len,fa,nxt[26];}T[N<<1]; //状态结点
14.
15. inline void SAM_extend(int c){ //在 SAM 中添加字符 c
16.     int np=++sz;
17.     T[np].len=T[lst].len+1;
18.     int x=lst;
19.     lst=np;
20.     while(x&&!T[x].nxt[c]) T[x].nxt[c]=np,x=T[x].fa;
21.     if(!x) T[np].fa=root;
22.     else{
23.         int y=T[x].nxt[c];
24.         if(T[y].len==T[x].len+1) T[np].fa=y;
25.         else{
26.             int nq=++sz;
27.             T[nq]=T[y];
28.             T[nq].len=T[x].len+1;
29.             T[nq].fa=T[y].fa;

```

```

29.         T[y].fa=T[np].fa=nq;
30.         while(x&&T[x].nxt[c]==y) T[x].nxt[c]=nq,x=T[x].fa;
31.     }
32. }
33. }
34. void SAM_init(){
35.     lst=sz=root=1;
36.     for(int i=1;i<=lenb;i++) SAM_extend(B[i]-'a');
37.     for(int i=1;i<=lenb;i++){
38.         int c=B[i]-'a';
39.         for(int j=i-1;j>=pre[c];j--) ch[j][c]=i;
40.         pre[c]=i;
41.     }
42.     for(int i=1;i<=sz;i++) sum[T[i].len]++;
43.     for(int i=1;i<=lenb;i++) sum[i]+=sum[i-1];
44.     for(int i=1;i<=sz;i++) mp[sum[T[i].len]--]=i;
45. }
46.
47. inline int trans(int x,int c,int y){
48.     if(!y)return T[x].nxt[c];
49.     else return ch[x][c];
50. }
51.
52. void Solve(int f1,int f2){
53.     memset(dp,inf,sizeof(dp));
54.     dp[f2^1]=0;
55.     int ans=inf;
56.     for(int i=1;i<=lena;i++){
57.         int c=A[i]-'a';
58.         for(int z=(f2?lenb:sz);z>=(f2^1);z--){
59.             int x=f2?z:mp[z];
60.             int y=trans(x,c,f2);
61.             if(!y) ans=min(ans,dp[x]+1);
62.             else{
63.                 dp[y]=min(dp[y],dp[x]+1);
64.                 if(!f1&&x!=(f2^1)) dp[x]=inf;
65.             }
66.         }
67.     }
68.     if(ans>=inf) puts("-1");
69.     else printf("%d\n",ans);
70. }
71.
72. signed main(){

```

```

73. // ios::sync_with_stdio(0);
74. // cin.tie(0);cout.tie(0);
75. // freopen("in.cpp","r",stdin);
76. // freopen("out.cpp","w",stdout);
77.     scanf("%s%s",A+1,B+1);
78.     lena=strlen(A+1);lenb=strlen(B+1);
79.     SAM_init();
80.     Solve(0,0);
81.     Solve(0,1);
82.     Solve(1,0);
83.     Solve(1,1);
84. //     finish=clock();
85. //     cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl;     return 0;
86. }

```

## （五）实验步骤与调试

HDU-2037 今年暑假不 AC

题目问最多选中多少个区间使得彼此不重叠。经典题，按右端点从小到大排序，然后贪心地往右选即可。可以证明这样选的区间数总是最多的（用弦图和区间图的知识证明）

HDU-1052 Tian Ji -- The Horse Racing

题目给定  $n$  匹马的战斗力的，问最多能赢多少局，进而算出总分。可以用双指针来做，把两人的马按战力从小到大排序，那么对于下等马和上等马，我们都需要尽量令其赢，如果两边都赢不了，那我们就选择用该下等马去博掉上等马。

LibreOJ-2123 最短不公共子串


SAM 的经典题吧应该。以  $B$  串建立后缀自动机，问题转化为最短的没有出现的字符串。在 SAM 上做 dp，设  $dp[i]$  表示到结点  $i$  时的最短长度。如果这个点右不是  $T$  中字符的出边则  $dp[i]=1$ , 否则  $dp[i]=\min(dp[j])+1$ 。

## （六）实验结果与分析

HDU-2037 今年暑假不 AC

结构体排一遍序直接遍历区间，判左端点和当前结束时间。


### #36673840 | LINNO's solution for [HDU-2037]

Status	Time	Memory	Length	Lang	Submitted	Open	Share text 	RemoteRunId
Accepted	15ms	1412kB	1062	G++	2022-06-14 17:30:34	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">37608979</a>

### HDU-1052 Tian Ji -- The Horse Racing

双指针分别指向田忌的下等马和上等马、王的下等马和上等马。然后判  $n$  轮的结果就行了。结果一共有 4 种，平局就是直接跳过。

### #36674662 | LINNO's solution for [HDU-1052]

Status	Time	Memory	Length	Lang	Submitted	Open	Share text 	RemoteRunId
Accepted	109ms	1436kB	1137	G++	2022-06-14 19:08:30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<a href="#">37609025</a>

### LibreOJ-2123 最短不公共子串

不懂贪心，板子套上去做。字符串永远的痛。

### #36677682 | LINNO's solution for [LibreOJ-2123]

Status	Time	Memory	Length	Lang
Accepted	192ms	6964kB	2167	G++ (ISO C++20) -O2 64bit

# 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     ℃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.          #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';ch=getchar();}return x*f;}
10.         // void write(int x){if(x>9) write(x/10);putch
   ar(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_S
   EC<<endl; return 0;
37.         }

```

## 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	

# 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 母函数 指导教师 张震  
实验项目编号 0806017906 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) 掌握母函数原理。
- 2) 学会如何使用母函数解决实际问题。

## (二) 实验内容和要求

实验内容::

1. HDU 1398
2. HDU 1028
3. LibreOJ-3114

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

## 1) HDU 1398 Square Coins

```
1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include <bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=1007;
8. const int mod=1e9+7;
9.
10. // 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;}
11. // void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
12.
13. int a[N],b[N],n;
14.
15. signed main(){
16.     ios::sync_with_stdio(0);
17.     cin.tie(0);cout.tie(0);
18. // freopen("in.cpp","r",stdin);
19. // freopen("out.cpp","w",stdout);
20.     int T=1;
21. // cin>>T;
22. // clock_t start,finish;
23. // start=clock();
24.     for(int i=0;i<=300;++i) a[i]=b[i]=0;
25.     a[0]=1;
26.     for(int i=1;i<=17;++i){ //一共有 17 种金币
27.         for(int j=0;j<=300;j+=i*i){ //b 的每一项系数
28.             for(int k=0;k<=300;++k){ //a 的每一项系数
29.                 b[k+j]+=a[k];
30.             }
31.         }
32.         for(int k=0;k<=300;++k){
33.             a[k]=b[k];
34.             b[k]=0;
35.         }
36.     }
37.     while(cin>>n){
38.         if(!n) break;
39.         else cout<<a[n]<<"\n";
40.     }
```



```

41. // finish=clock();
42. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
43. }

```

## 2) HDU 1028 Ignatius and the Princess III

```

1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include <bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=2e5+7;
8. const int mod=1e9+7;
9.
10. // 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';ch=getchar();}return x*f;}
11. // void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
12.
13. int n,dp[200][200];
14.
15. signed main(){
16. // ios::sync_with_stdio(0);
17. // cin.tie(0);cout.tie(0);
18. // freopen("in.cpp","r",stdin);
19. // freopen("out.cpp","w",stdout);
20. int T=1;
21. // cin>>T;
22. // clock_t start,finish;
23. // start=clock();
24. for(int i=1;i<=120;++i) dp[i][1]=dp[1][i]=dp[0][i]=1;
25. for(int i=2;i<=120;++i){
26.     for(int j=2;j<=i;++j) dp[i][j]=dp[i][j-1]+dp[i-j][j];
27.     for(int j=i+1;j<=120;++j) dp[i][j]=dp[i][i];
28. }
29. while(cin>>n){
30.     cout<<dp[n][n]<<"\n";
31. }
32. // finish=clock();
33. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl; return 0;
34. }

```

## 3) LibreOJ-3114 移动金币

```

1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include <bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=2e5+7,M=55;
8. const int mod=1e9+9;
9.
10. 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';ch=
    getchar();}return x*f;}
11. void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
12.
13. int dp[22][N],n,m,num,tot,ou,frac[N+M],ifrac[N+M];
14.
15. inline int fpow(int a,int b){
16.     int res=1;
17.     while(b){
18.         if(b&1) res=res*a%mod;
19.         a=a*a%mod;
20.         b>>=1;
21.     }
22.     return res;
23. }
24.
25. inline int inv(int x){
26.     return fpow(x,mod-2);
27. }
28.
29. inline void init(){
30.     int len=150050;
31.     frac[0]=ifrac[0]=1;
32.     for(int i=1;i<=len;i++) frac[i]=frac[i-1]*i%mod;
33.     ifrac[len]=inv(frac[len]);
34.     for(int i=len-1;i;i--) ifrac[i]=ifrac[i+1]*(i+1)%mod;
35. }
36.
37. inline int C(int n,int m){
38.     if(n<m) return 0;
39.     return frac[n]*ifrac[m]%mod*ifrac[n-m]%mod;
40. }
41.
42. signed main(){

```

```

43.     ios::sync_with_stdio(0);
44.     cin.tie(0);cout.tie(0);
45. //   freopen("in.cpp","r",stdin);
46. //   freopen("out.cpp","w",stdout);
47. //   cin>>T;
48. //   clock_t start,finish;
49. //   start=clock();
50.     init();
51.     n=read(),m=read();
52.     num=(m+1)/2,ou=m-num+1,tot=n-m;
53.     dp[20][tot]=1;
54.     for(int i=19;~i;i--)
55.         for(int j=0;j<=tot;j++){
56.             for(int k=0;j+(1<<i)*k<=tot&&k<=num;k+=2){
57.                 dp[i][j]=(dp[i][j]+dp[i+1][j+(1<<i)*k]*C(num,k)%mod)%mod;
58.             }
59.         }
60.     int res=0;
61.     for(int j=0;j<=tot;j++){
62.         res=(res+dp[0][j]*C(j+ou-1,ou-1)%mod)%mod;
63.     }
64.     cout<<(C(n,m)-res+mod)%mod<<"\n";
65. //   finish=clock();
66. //   cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl;
67.     return 0;
68. }

```

## （五）实验步骤与调试

### 1) HDU 1398 Square Coins

母函数裸题，令  $X=(1+x+x^2+\dots+x^{289})(1+x^4+x^8+\dots+x^{288})\dots(1+x^{17}+\dots+x^{289})$ ，然后项数相乘，系数相加，直接转移一下，最后得到  $x^n$  的系数就是答案。

### 2) HDU 1028 Ignatius and the Princess III

直接上 DP， $dp[i][j]$  表示最大数为  $j$  时  $i$  的拆分方案数，那么状态可以由  $dp[i][j-1]$  和  $dp[i-j][j]$  转移得来。

### 3) LibreOJ-3114 移动金币

阶梯 Nim，先手必败的充要条件是奇数位置异或和为 0， $dp[i][j]$  表示前  $i$  为剩下  $j$  石子时异或和为 0 的方案数，枚举、插板就完了。

## （六）实验结果与分析

### 1) HDU 1398 Square Coins

[#36679762](#) | [LINNO's solution for \[HDU-1398\]](#)

Status	Memory	Length	Lang	Submitted	Open	Share text ?
Accepted	1440kB	1072	G++	2022-06-15 11:09:52	<input checked="" type="checkbox"/>	<input type="checkbox"/>

### 2) HDU 1028 Ignatius and the Princess III

母函数的解法没写，不过差不多长度吧。

[#36679557](#) | [LINNO's solution for \[HDU-1028\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	15ms	1592kB	999	G++	2022-06-15 10:33:01	<input checked="" type="checkbox"/>

### 3) LibreOJ-3114 移动金币

[#36680034](#) | [LINNO's solution for \[LibreOJ-3114\]](#)

Status	Time	Memory	Length	Lang
Accepted	1388ms	28604kB	1605	G++ (ISO C++20) -O2 64bit

# 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 搜索算法 指导教师 张震  
实验项目编号 0806017907 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) 掌握深度优先搜索算法。
- 2) 能够利用剪枝算法提高搜索效率。

## （二）实验内容和要求

实验内容::

1. HDU 1238
2. HDU 1010
3. LibreOJ-10029
4. LibreOJ-10249

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

## 1) HDU 1238 Substrings

```
1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include <bits/stdc++.h>
4. using namespace std;
5. const int N=207;
6. const int mod=1e9+7;
7.
8. // 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;}
9. // void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
10.
11. string s[N];
12. int n;
13.
14. void Solve(){
15.     cin>>n;
16.     for(int i=1;i<=n;++i) cin>>s[i];
17.     s[0]=s[1];
18.     reverse(s[0].begin(),s[0].end());
19.     int ans=0,LEN=s[1].length();
20.     for(int i=LEN;i>=1;--i){
21.         for(int st=0;st+i-1<LEN;++st){
22.             string s1=s[1].substr(st,i);
23.             string s2=s[0].substr(st,i);
24.             int f1=1,f2=1;
25.             for(int j=2;j<=n;++j){
26.                 if(s[j].find(s1)==s[j].npos) f1=0;
27.                 if(s[j].find(s2)==s[j].npos) f2=0;
28.             }
29.             // cout<<i<<" "<<f1<<" "<<f2<<" "<<s1<<" "<<s2<<"!!\n";
30.             if(f1||f2){
31.                 ans=i;
32.                 break;
33.             }
34.         }
35.         if(ans) break;
36.     }
37.     cout<<ans<<"\n";
38. }
39.
40. signed main(){
```

```

41. ios::sync_with_stdio(0);
42. cin.tie(0);cout.tie(0);
43. // freopen("in.cpp","r",stdin);
44. // freopen("out.cpp","w",stdout);
45. int T=1;
46. cin>>T;
47. // clock_t start,finish;
48. // start=clock();
49. while(T--){
50.     Solve();
51. }
52. // finish=clock();
53. // cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl;
54. return 0;
55. }

```

## 2) HDU 1010 Tempter of the Bone

```

1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. using namespace std;
5.
6. 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';ch=ge
    tchar();}return x*f;}
7. void write(int x){if(x>9) write(x/10);putchar(x%10+'0');}
8.
9. const int xx[]={0,0,1,-1},yy[]={1,-1,0,0};
10. int n,m,t,sx,sy,dx,dy,flag;
11. int ex[15][15];
12. char mp[15][15];
13.
14. inline int Abs(int x){return x<0?-x:x;}
15.
16. inline void dfs(int x,int y,int lf){
17.     if(x==dx&&y==dy&&lf==0) flag=1;
18.     int dis=lf-Abs(dx-x)-Abs(dy-y);
19.     if(flag || dis<0 || (dis&1)) return;
20.     for(int d=0;d<4;++d){
21.         int nx=x+xx[d],ny=y+yy[d];
22.         if(nx<1 || ny<1 || nx>n || ny>m || mp[nx][ny]=='X' || ex[nx][ny]) conti
nue;

```

```

23.         ex[nx][ny]=1;
24.         dfs(nx,ny,lf-1);
25.         ex[nx][ny]=0;
26.     }
27. }
28.
29. void Solve(){
30.     memset(ex,0,sizeof(ex));
31.     flag=0;
32.     for(int i=1;i<=n;++i){
33.         for(int j=1;j<=m;++j){
34.             cin>>mp[i][j];
35.             if(mp[i][j]=='S') sx=i,sy=j;
36.             if(mp[i][j]=='D') dx=i,dy=j;
37.         }
38.     }
39.     ex[sx][sy]=1;
40.     dfs(sx,sy,t);
41.     if(flag) cout<<"YES\n";
42.     else cout<<"NO\n";
43. }
44.
45. signed main(){
46. //  freopen("in.cpp","r",stdin);
47. //  freopen("out.cpp","w",stdout);
48. //  int T=1;
49. //  cin>>T;
50. //  clock_t start,finish;
51. //  start=clock();
52. while(cin>>n>>m>>t){
53.     if(!n&&!m&&!t) break;
54.     Solve();
55. }
56. //  finish=clock();
57. //  cerr<<((double)finish-start)/CLOCKS_PER_SEC<<endl;
58. return 0;
59. }

```

### 3) LibreOJ-10029 棋盘游戏

```

1. // #pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. // #pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. using namespace std;
5. const int N=2e5+7;

```



```

6. const int mod=1e9+7;
7.
8. //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;}
9.
10. struct X{
11.     string a[4];
12.     int step;
13. }s,t;
14.
15. int ans=1e7;
16. int xx[]={1,0,-1,0},yy[]={0,1,0,-1};
17. queue<X>q;
18. map<string,int>vis;
19.
20. string get(X x){
21.     string s="";
22.     for(int i=0;i<4;i++) s+=x.a[i];
23.     return s;
24. }
25.
26. signed main(){
27.     ios::sync_with_stdio(0);
28.     cin.tie(0);cout.tie(0);
29.     for(int i=0;i<4;i++){
30.         cin>>s.a[i];
31.         s.step=1;
32.     }
33.     for(int i=0;i<4;i++){
34.         cin>>t.a[i];
35.     }
36. // cout<<get(s)<<"\n";
37.     q.push(s);
38.     while(!q.empty()){
39.         X fro=q.front();
40.         q.pop();
41.         if(vis[get(fro)]) continue;
42.         vis[get(fro)]=fro.step;
43.         for(int i=0;i<4;i++){
44.             for(int j=0;j<4;j++){
45.                 if(fro.a[i][j]!=t.a[i][j]){
46.                     for(int d=0;d<4;d++){
47.                         int nx=i+xx[d],ny=j+yy[d];

```

```

48.             if(nx>=0&&nx<4&&ny>=0&&ny<4){
49.                 swap(fro.a[i][j],fro.a[nx][ny]);
50.                 fro.step++;
51.                 q.push(fro);
52.                 swap(fro.a[i][j],fro.a[nx][ny]);
53.                 fro.step--;
54.             }
55.         }
56.     }
57. }
58. }
59. }
60. cout<<vis[get(t)]-1<<"\n";
61. return 0;
62. }

```

#### 4) LibreOJ-10249

```

1. //pragma GCC optimize("Ofast", "inline", "-ffast-math")
2. //pragma GCC target("avx,sse2,sse3,sse4,mmx")
3. #include<bits/stdc++.h>
4. #define inf 0x3f3f3f3f
5. #define int long long
6. using namespace std;
7. const int N=5e5+7;
8. const int mod=1e9+7;
9.
10. //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;}
11. int n,m,sum=0,flag=0,a[N];
12. int ans[N];
13.
14. vector<int>vis(N);
15.
16. void dfs(int pos,int l,int r,int pre,int suf){
17.     if(flag) return;
18.     if(!vis[a[pos]-pre]&&!vis[a[pos]-suf]) return;
19.     if(l==r){ //取完 n 个了
20.         ans[l]=a[2*n]-pre-suf; //这是最后一个数
21.         if(ans[l]<1||ans[l]>500) return;
22.         for(int i=1;i<=n;i++) cout<<ans[i]<<" ";
23.         flag=1;
24.         return;
25.     }

```

```

26.     if(vis[a[pos]-pre]){ //顺着来取
27.         ans[l]=a[pos]-pre;
28.         dfs(pos+1,l+1,r,a[pos],suf);
29.     }
30.     if(vis[a[pos]-suf]){ //逆着来取
31.         ans[r]=a[pos]-suf;
32.         dfs(pos+1,l,r-1,pre,a[pos]);
33.     }
34. }
35.
36. signed main(){
37.     ios::sync_with_stdio(0);
38.     cin.tie(0);cout.tie(0);
39.     cin>>n;
40.     for(int i=1;i<=2*n;i++) cin>>a[i];
41.     sort(a+1,a+1+2*n);
42.     cin>>m;
43.     for(int i=1,x;i<=m;i++){
44.         cin>>x;
45.         vis[x]=1;
46.     }
47.     dfs(1,1,n,0,0);
48.     return 0;
49. }

```

## （五）实验步骤与调试

### 1) HDU 1238 Substrings

要求找到一个子串，使得在所有（字符串或其反串）都出现过。使用字符串的 `find` 操作，遍历  $2 \sim n$  所有字符串，对 `s1` 和 `s1` 的反串的所有子串判一下就完事了。

### 2) HDU 1010 Tempter of the Bone

深搜剪一下枝。首先为了能达到恰好 `t` 时间到达也许会绕一下圈嘛，这里可以用曼哈顿距离的奇偶性剪一下，然后就是正常做。

### 3) LibreOJ-10029 棋盘游戏

直接 BFS。用一个 `map` 记录棋盘的状态以及他需要的步数即可。

### 4) LibreOJ-10249 Weight

这道题要求剪枝力度比较足，不过他要搜的数是非常有限的，我们记一下出现的前缀和以及后缀和，然后可以记录根据他们的出现次数来 `dfs`，最好双向，这样可以对每个数都找一遍，然后判断正放还是反放。单向的话要搜比较多的数，不太好剪。

## （六）实验结果与分析

### 1) HDU 1238 Substrings

直接找子串完事了。

[#36680779](#) | [LINNO's solution for \[HDU-1238\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	140ms	1480kB	1272	G++	2022-06-15 13:49:42	<input checked="" type="checkbox"/>

### 2) HDU 1010 Tempter of the Bone

这题的输入真的有毒……WA 了半天。

[#36681430](#) | [LINNO's solution for \[HDU-1010\]](#)

Status	Time	Memory	Length	Lang	Submitted	Open
Accepted	296ms	1408kB	1391	G++	2022-06-15 15:06:26	<input checked="" type="checkbox"/>

### 3) LibreOJ-10029 棋盘游戏

没啥好注意的。

[#36681467](#) | [LINNO's solution for \[LibreOJ-10029\]](#)

Status	Time	Memory	Length	Lang
Accepted	300ms	8716kB	1247	G++ (ISO C++20) -O2 64bit

### 4) LibreOJ-10249

搞定！

[#36681548](#) | [LINNO's solution for \[LibreOJ-10249\]](#)

Status	Time	Memory	Length	Lang
Accepted	47ms	6448kB	1125	G++ (ISO C++20) -O2 64bit

# 暨南大学本科实验报告专用纸(附页)

---