

Volleyball Match - Week 1 Editorial

Apr 23 2014

Difficulty level : Easy

Required knowledge : Combinatorics/DP

Problem setter : [Sergey Kulik](#)

Problem tester : [Wanbo](#)

Approach

At first, there are a few tricky cases that should be considered. The answer in each of these cases is 0

- $A = 24$, $B = 25$, or vice versa. The game hasn't ended in this case.
- The both A and B are less than 25.
- $A = B$
- One of the teams's scores is greater than 25 and the difference between the teams' scores is different from 2.

After checking these cases, let's calculate a number of games that has the *intermediary* score of X : Y over all the possible pairs (X, Y) where the both X and Y are not greater than 25. Let's name it $S(X, Y)$. By *intermediary* we mean the the game doesn't have to be ended with this score (but this score CAN correspond to some ended game, as well).

We can calculate $S(X, Y)$ using the recurrent formula: $S(X, Y) = S(X - 1, Y) + S(X, Y - 1)$ in the general case, starting with $S(0, 0) = 1$. As we can see, it's quite similar to the Pascal Triangle, so we can use the number of combinations instead of the recurrent formula. Nevertheless, the writer of the problem considers recurrent formula the simplest solution.

Now, if both A and B aren't greater than 25, we can just output the value of $S(A-1, B)$ or $S(A, B-1)$ (depending on who is the winner of the game) modulo $10^9 + 7$.

What happens if A or B (or both) are greater than 25. That means the the score of 24 : 24 was reached during the game. Then, the score of 25 : 25 was also reached. Otherwise it would mean that the game ends with the score of 24 : 26 or 26 : 24. There are two ways to reach 25 : 25 from 24 : 24. Then, the score of 26 : 26 was also reached, and again there're two ways to go from 25 : 25 to 26 : 26. This goes on, till the score of $\min(A, B) : \min(A, B)$ is reached. Then, one of teams gains two points in a row and the game ends. There are always two ways to get $C : C$ from $C-1 : C-1$.

So, according to the simplest combinatorial rules, if $\max(A, B)$ is greater than 25, the answer is $S(24, 24) * 2^{\min(A, B) - 24}$.

Setter's code

```
#include <cstdio>
#include <algorithm>
using namespace std;

int n, m, i, j, ret, dp[26][26];

int pw (int k) {
    if (!k) return 1;
    if (k == 1) return 2;
    int q = pw(k / 2);
    q = (1LL * q * q) % 1000000007;
    if (k % 2) return (q + q) % 1000000007; else return q;
}
```

```

int main (int argc, char * const argv[]) {
    scanf("%d %d", &n, &m);
    if ((n < 25 && m < 25) || ((n > 25 || m > 25) && abs(n - m) != 2) || (n
== 24 && m == 25) || (n == 25 && m == 24) || (n == m)) {
        printf("0\n");
        return 0;
    }
    dp[0][0] = 1;
    for(i = 0; i <= 25; i++) for(j = 0; j <= 25; j++) if (i + j) {
        if (i) dp[i][j] += dp[i - 1][j];
        if (j) dp[i][j] += dp[i][j - 1];
        dp[i][j] %= 1000000007;
    }
    if (n < m) swap(n, m);
    if ((n == 25 && n > m)) {
        printf("%d\n", dp[n - 1][m]);
        return 0;
    }
    ret = dp[24][24];
    ret = (1LL * ret * pw(min(n, m) - 24)) % 1000000007;
    printf("%d\n", ret);
    return 0;
}

```

Tester's Code

```

#include <map>
#include <set>
#include <list>
#include <queue>
#include <deque>
#include <stack>
#include <bitset>
#include <vector>
#include <ctime>
#include <cmath>
#include <cstdio>
#include <string>
#include <cstring>
#include <cassert>
#include <numeric>
#include <iomanip>
#include <sstream>
#include <fstream>
#include <iostream>
#include <algorithm>
using namespace std;
typedef long long LL;
typedef pair<int, int> PII;
typedef vector<int> VI;
typedef vector<LL> VL;
typedef vector<PII> VPII;
#define MM(a,x) memset(a,x,sizeof(a));
#define ALL(x) (x).begin(), (x).end()
#define P(x) cerr<<"[#x<<" = "<<(x)<<"\n"
#define PP(x,i) cerr<<"[#x<<i<<" = "<<x[i]<<"\n"
#define P2(x,y) cerr<<"[#x" = "<<(x)<<", "#y" = "<<(y)<<"\n"
#define TM(a,b) cerr<<"[#a" -> "#b": "<<1e3*(b-a)/CLOCKS_PER_SEC<<"ms]\n";
#define UN(v) sort(ALL(v)), v.resize(unique(ALL(v))-v.begin())
#define mp make_pair
#define pb push_back
#define x first

```

```

#define y second
struct _ {_() {ios_base::sync_with_stdio(0);}} _;
template<class A, class B> ostream& operator<< (ostream &o, pair<A, B> t) {o <<
 "(" << t.x << ", " << t.y << ")"; return o;}
template<class T> void PV(T a, T b) {while(a != b)cout << *a++, cout << (a !=
 b ? " " : "\n");}
template<class T> inline bool chmin(T &a, T b) {return a > b ? a = b, 1 : 0;}
template<class T> inline bool chmax(T &a, T b) {return a < b ? a = b, 1 : 0;}
template<class T> string toString(T x, int len = 0) {stringstream ss; ss << x;
 string r = ss.str(); if(r.length() < len) r = string(len - r.length(), '0') + r;
 return r;}
template<class T> void convert(string x, T& r) {stringstream ss(x); ss >> r;}
const int inf = ~0u >> 2;
const long long linf = ~0ull >> 2;
const int mod = int(1e9) + 7;
const int N = 111111;

```

```

int dp[30][30];

```

```

LL f(int A, int B) {
    int &r = dp[A][B];
    if(r == -1) {
        if(A == 0 || B == 0) r = 1;
        else r = (f(A - 1, B) + f(A, B - 1)) % mod;
    }
    return r;
}

```

```

LL exp(LL x, LL n, LL m = ~0ULL >> 1) {
    x %= m;
    LL r = 1 % m;
    while(n) {
        if(n & 1) r = r * x % m;
        x = x * x % m, n >>= 1;
    }
    return r;
}

```

```

int main() {
    int A, B;
    cin >> A >> B;
    MM(dp, -1);
    if(A > B) swap(A, B);
    if(B < 25 || B - A < 2 || (B > 25 && B - A != 2)) cout << 0 << endl;
    else if(A >= 24) {
        cout << (LL) exp(2, A - 24, mod) * f(24, 24) % mod << endl;
    } else {
        B--;
        cout << f(A, B) << endl;
    }
    return 0;
}

```