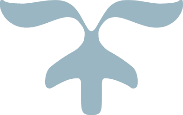


DAA WEEK – 11 SKILL – 11



# [Kingdom Division](https://www.hackerrank.com/contests/daa-skill-11-dynamic-programming-part-ii/challenges/kingdom-division)

#include <stdio.h>

#include <stdlib.h>

#define MOD 1000000007

typedef struct \_lnode{

int x;

int w;

struct \_lnode \*next;

} lnode;

void insert\_edge(int x,int y,int w);

void dfs(int x,int y);

long long not\_care[100000],safe[100000];

lnode \*table[100000]={0};

int main(){

int n,x,y,i;

scanf("%d",&n);

for(i=0;i<n-1;i++){

scanf("%d%d",&x,&y);

insert\_edge(x-1,y-1,0);

}

dfs(0,-1);

printf("%lld",safe[0]\*2%MOD);

return 0;

}

void insert\_edge(int x,int y,int w){

lnode \*t=(lnode\*)malloc(sizeof(lnode));

t->x=y;

t->w=w;

t->next=table[x];

table[x]=t;

t=(lnode\*)malloc(sizeof(lnode));

t->x=x;

t->w=w;

t->next=table[y];

table[y]=t;

return;

}

void dfs(int x,int y){

int f=0;

long long not\_safe=1;

lnode \*p;

not\_care[x]=1;

for(p=table[x];p;p=p->next)

if(p->x!=y){

dfs(p->x,x);

f=1;

not\_care[x]=(not\_care[p->x]+safe[p->x])%MOD\*not\_care[x]%MOD;

not\_safe=not\_safe\*safe[p->x]%MOD;

}

if(!f)

safe[x]=0;

else

safe[x]=(not\_care[x]-not\_safe+MOD)%MOD;

return;

}

**Kingdom Division Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Sam and substrings](https://www.hackerrank.com/contests/daa-skill-11-dynamic-programming-part-ii/challenges/sam-and-substrings)

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAXLEN 200001

#define MODULUS 1000000007LL

int main() {

long long total, multiplier, partialsum;

char c;

total = partialsum = 0;

multiplier = 1;

while (1) {

c=getchar();

if (c<'0' || c>'9') break;

c -= '0';

partialsum += multiplier \* c;

total = (total \* 10 + partialsum) % MODULUS;

multiplier++;

}

printf("%lld\n",total);

return 0;

}

**Sam and substrings Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Sherlock and Cost](https://www.hackerrank.com/contests/daa-skill-10-dynamic-programming-part-i/challenges/sherlock-and-cost)Fibonacci Modified

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#define MAXL 26624

unsigned int MADD(unsigned int\* pC, unsigned int\* pB, unsigned int\* pA, unsigned int n)

{

unsigned int i,j,x;

for (i=0; i<n; i++) pC[i] = pA[i];

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

{

for (j=0; j<n; j++)

{

if ((x = (pC[i+j] += pB[i]\*pB[j])) < 10000) continue;

x /= 10000; pC[i+j+1] += x; pC[i+j] -= x\*10000;

}

if ((x = pC[i+j]) < 10000) continue;

x /= 10000; pC[i+j+1] += x; pC[i+j] -= x\*10000;

}

n <<= 1; while (pC[n-1] == 0) n--;

return n;

}

int main() {

/\* Enter your code here. Read input from STDIN. Print output to STDOUT \*/

unsigned int N,m,n=1;

unsigned int\* p;

unsigned int\* pA;

unsigned int\* pB;

unsigned int\* pC;

unsigned int A[MAXL];

unsigned int B[MAXL];

unsigned int C[MAXL];

memset(pA=A, 0, sizeof(A));

memset(pB=B, 0, sizeof(B));

memset(pC=C, 0, sizeof(C));

scanf("%d %d %d\n", pA, pB, &N);

while (N-- > 2)

{

n = MADD(pC, pB, pA, n);

p = pC; pC=pA; pA = pB; pB=p;

}

printf("%d", p[--n]);

while (n > 0) printf("%04u", p[--n]);

printf("\n");

return 0;

}

**Fibonacci Modified Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

# [Abbreviation](https://www.hackerrank.com/contests/daa-skill-11-dynamic-programming-part-ii/challenges/abbr)

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int f(char \*a, int i, char \*b, int j) {

if (a[i] == '\0' && b[j] == '\0')

return 1;

if (a[i] == '\0')

return 0;

if (b[j] == '\0') {

while (a[i] != '\0') {

if (a[i] <= 'Z' && a[i] >= 'A')

return 0;

i++;

}

return 1;

}

if ((a[i] <= 'Z' && a[i] >= 'A') && (a[i] != b[j]))

return 0;

if (a[i] == b[j])

return f(a, i + 1, b, j + 1);

if ((a[i] - 'a' + 'A') == b[j]) {

if (f(a, i + 1, b, j + 1))

return 1;

else

return f(a, i + 1, b, j);

}

return f(a, i + 1, b, j);

}

int main() {

char a[1010], b[1010];

int q;

scanf("%d", &q);

while (q--) {

scanf("%s", a);

scanf("%s", b);

if (f(a, 0, b, 0)) {

printf("YES\n");

} else {

printf("NO\n");

}

}

return 0;

}

**Abbreviation Test Cases**

**A screenshot of a computer

AI-generated content may be incorrect.**

**SKILL WEEK – 10**

[**https://www.hackerrank.com/contests/daa-skill-11-dynamic-programming-part-ii/challenges**](https://www.hackerrank.com/contests/daa-skill-11-dynamic-programming-part-ii/challenges)