Fibonacci Matrix Exponential

/\*\*

f(n) = f(n-1) + f(n-2);

\*/

vector< vector<int> > a{{1, 1}, {1, 0}}, b{{1}, {0}};

vector< vector<int> > matrixMul(vector< vector<int> > a, vector< vector<int> > b){

if(a[0].size() != b.size()) return vector< vector<int> >(1, vector<int>(1, -1));

int n = a.size();

int m = b.size();

int p = b[0].size();

vector< vector<int> > ans(n, vector<int>(p, 0));

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

for(int j=0; j<p; j++)

for(int k=0; k<m; k++)

ans[i][j] += (a[i][k] \* b[k][j]);

return ans;

}

vector< vector<int> > powerMat(vector< vector<int> > vec, int p){

if(p == 1) return vec;

vector< vector<int> > ans = powerMat(vec, p>>1);

ans = matrixMul(ans, ans);

if(p & 1) ans = matrixMul(ans, vec);

return ans;

}

int findNFib(int n){

if(n <= 2) return 1;

return matrixMul(powerMat(a, n-1), b)[0][0];

}

void display(vector< vector<int> > vec){

for(auto x:vec){

for(auto y:x)

cout<<y<<" ";

cout<<endl;

}

}

int main()

{

/// inOut;

for(int i=1; i<=20; i++)

cout<<findNFib(i)<<endl;

return 0;

}

Example 01:

/\*\*

f(n) = a \* f(n-1) + b \* f(n-2) + c \* f(n-3);

\*/

vector< vector<int> > aa, bb{{1}, {1}, {0}};

vector< vector<int> > matrixMul(vector< vector<int> > a, vector< vector<int> > b){

if(a[0].size() != b.size()) return vector< vector<int> >(1, vector<int>(1, -1));

int n = a.size();

int m = b.size();

int p = b[0].size();

vector< vector<int> > ans(n, vector<int>(p, 0));

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

for(int j=0; j<p; j++)

for(int k=0; k<m; k++)

ans[i][j] += (a[i][k] \* b[k][j]);

return ans;

}

vector< vector<int> > powerMat(vector< vector<int> > vec, int p){

if(p == 1) return vec;

vector< vector<int> > ans = powerMat(vec, p>>1);

ans = matrixMul(ans, ans);

if(p & 1) ans = matrixMul(ans, vec);

return ans;

}

int findNth(int n){

if(n<3) return 1;

return matrixMul(powerMat(aa, n-2), bb)[0][0];

}

int main()

{

/// inOut;

int a, b, c; sfiii(a, b, c);

aa.push\_back({a, b, c});

aa.push\_back({1, 0, 0});

aa.push\_back({0, 1, 0});

for(int i=1; i<=10; i++)

cout<<findNth(i)<<endl;

return 0;

}