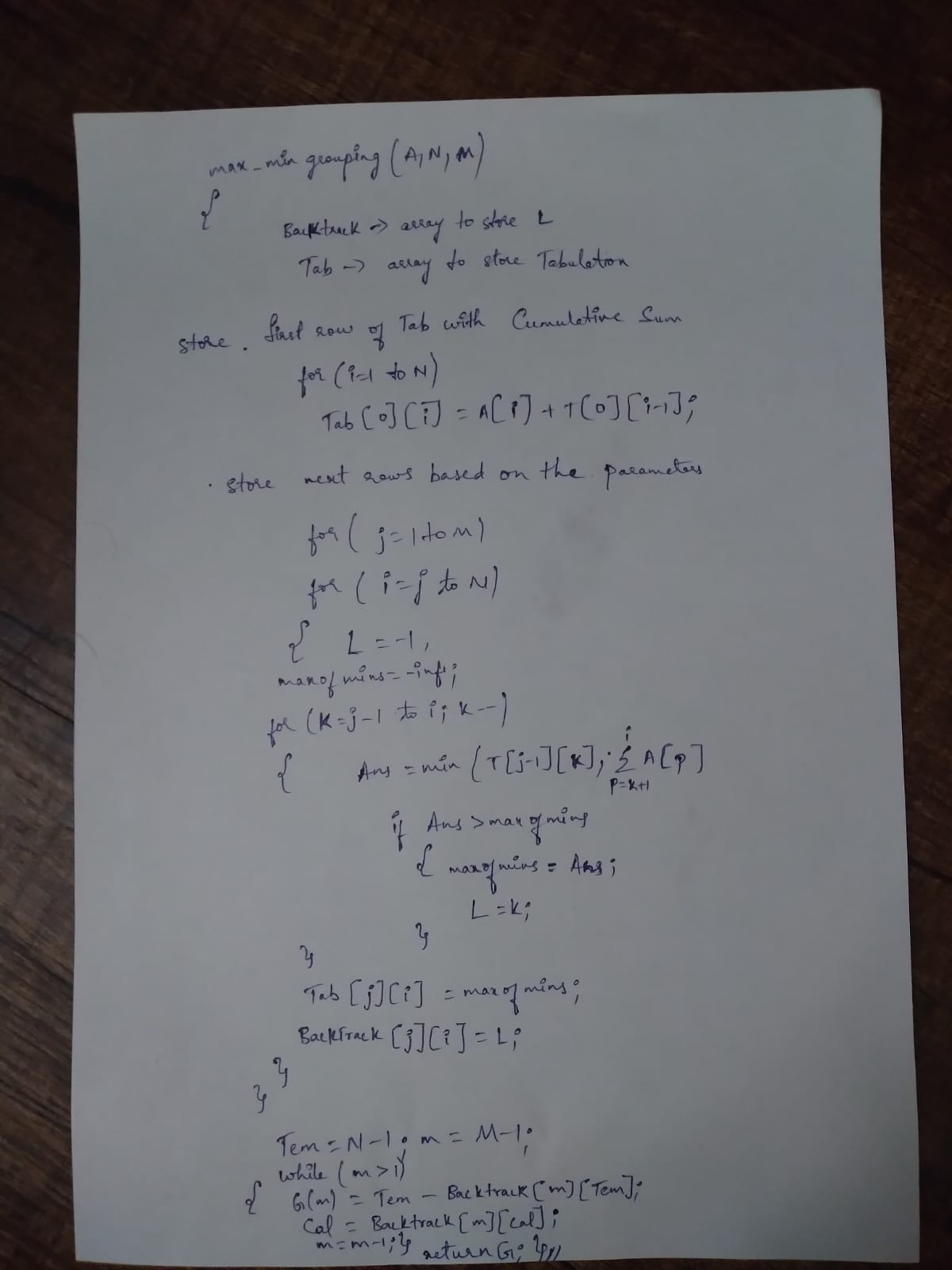
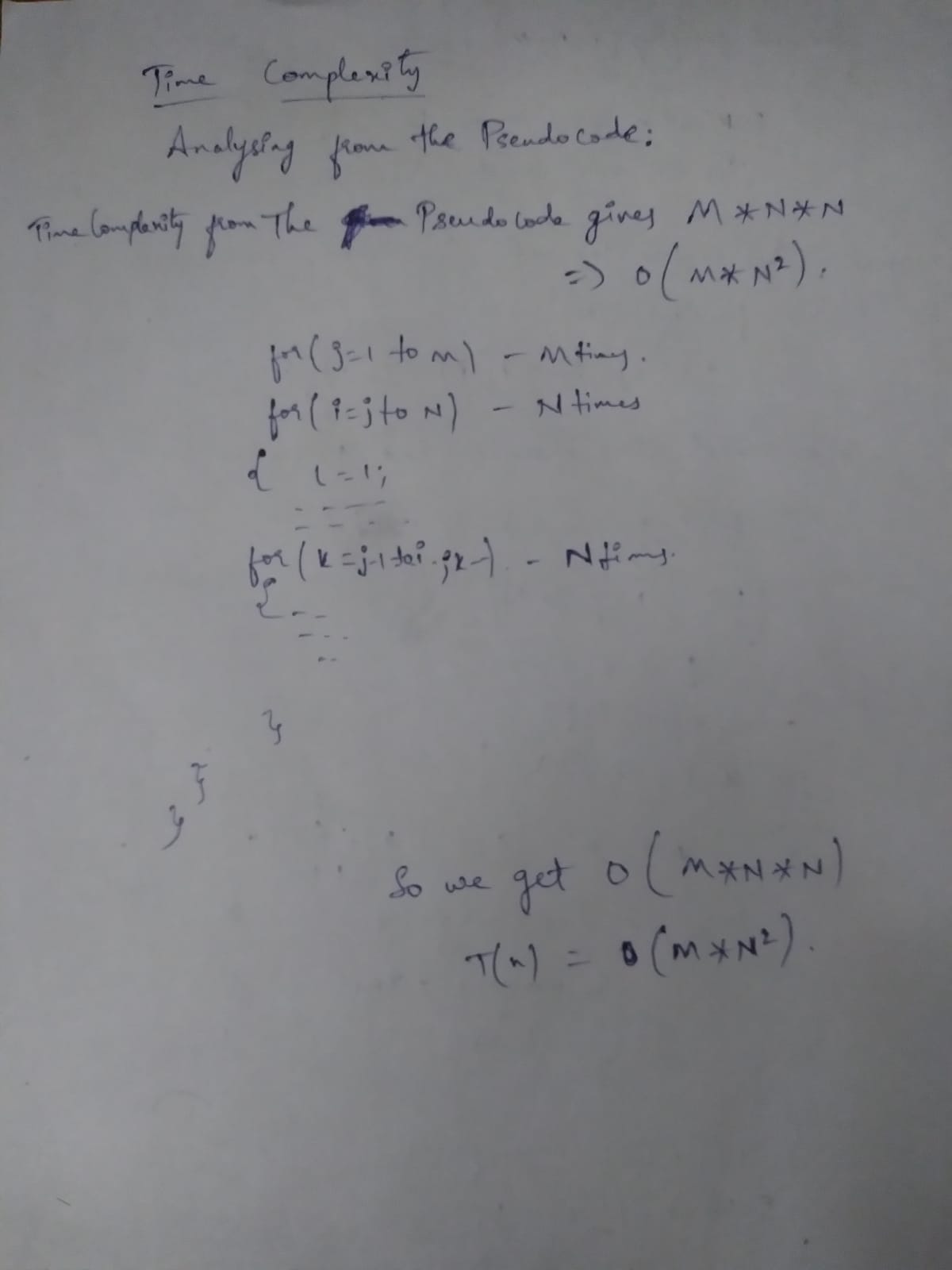
1)

Pseudocode for the dynamic program is in the below image



2)

Analysis of the running time asymptotically is in the below image



3) Grouping results of several input

ARRAY:-3 9 7 8 2 6 5 10 1 7 6 4

GROUPING FOR M-->3: 3 4 5

ARRAY:-3 9 7 8 2 6 5 10 1 7 6 4

GROUPING FOR M-->3: 3 4 5

ARRAY:-3 9 7 8 2 6 5 10 1 7 6 4

GROUPING FOR M-->5: 2 2 3 2 3

ARRAY:-3 9 7 5 2 6 8 17 6 4

GROUPING FOR M-->3: 3 4 3

ARRAY:-3 9 7 5 2

GROUPING FOR M-->3: 2 1 2

ARRAY:-3 9 7 5 2

GROUPING FOR M-->2: 2 3

4)SOURCE CODE

#include <stdio.h>

#include <iostream>

#include <vector>

using namespace std;

int sum(vector<int> A,int start ,int stop)

{

int sum=0;

for (int i=start;i<=stop;i++)

{

sum=sum+A[i];

}

return sum;

}

void printarray(vector<int> v)

{

for (int i=0;i<v.size();i++)

{

cout<<v[i]<<" ";

}

cout<<endl;

}

vector<int> Max\_min\_grouping(vector<int> A,int N,int M)

{

vector<vector<int>> backtrack(M, vector<int>(N, -1));

vector<vector<int>> table(M, vector<int>(N, 0));

table[0][0]=A[0];

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

{

table[0][i]=A[i]+table[0][i-1];

}

for (int j=1;j<M;j++)

{

for (int i=j;i<N;i++)

{

int c=-1;

int best\_param=-1000;

for (int k=j-1;k<i;k++)

{

int val=min(table[j-1][k],sum(A,k+1,i));

if (val>best\_param)

{

best\_param=val;

c=k;

}

}

table[j][i]=best\_param;

backtrack[j][i]=c;

}

}

vector<int> G(M, 0);

int col\_mark=N-1;

int m=M;

while(m>0)

{

m=m-1;

G[m]=col\_mark-backtrack[m][col\_mark];

col\_mark=backtrack[m][col\_mark];

}

return (G);

}

int main()

{

vector<int> A;

int M,N;

A = {3,9,7,8,2,6,5,10,1,7,6,4};

N = 12;

M = 3;

cout <<"ARRAY";

printarray(A);

vector<int> G=Max\_min\_grouping(A,N,M);

cout <<"GROUPING FOR M-->"<<M<<": ";

printarray(G);

A = {3,9,7,8,2,6,5,10,1,7,6,4};

N = 12;

M = 5;

cout <<"ARRAY";

printarray(A);

cout <<"GROUPING FOR M-->"<<M<<": ";

printarray(Max\_min\_grouping(A,N,M));

A = {3,9,7,5,2,6,8,17,6,4};

N = 10;

M = 3;

cout <<"ARRAY";

printarray(A);

cout <<"GROUPING FOR M-->"<<M<<": ";

printarray(Max\_min\_grouping(A,N,M));

A = {3,9,7,5,2};

N = 5;

M = 3;

cout <<"ARRAY";

printarray(A);

cout <<"GROUPING FOR M-->"<<M<<": ";

printarray(Max\_min\_grouping(A,N,M));

A = {3,9,7,5,2};

N = 5;

M = 2;

cout <<"ARRAY";

printarray(A);

cout <<"GROUPING FOR M-->"<<M<<": ";

printarray(Max\_min\_grouping(A,N,M));

return 0;

}