CREATING A TREE WITH AN ARRAY

->*Disclaimer: Code illustrations are in C++ Syntax*

Initialize an integer, n=0

Initialize a depth tracker, depthCounter=2

Prompt the user to enter the depth of the tree and validate it i.e. (n>0)

Initialize an array with size[2n-1], to 0s i.e. tree\_array[((int)pow(2,n)-1)]={0}

Prompt the user for the value of the root node. Store it in the array, at array index 1

Check if n=1 and skip the loop below

Define an infinite loop with a counter c, incrementing c by 1 in each loop

Initialize a variable to keep track of the previous value of the loop counter to 2, prevC=2

In the loop:

Check if the user has finished entering the values of a certain depth

Check if n is equal to depthCounter. It true, break the loop

Double prevC

Increment depthCounter by 1

Ie

if((2\*c)==(2\*prevC)){

if(\*nPtr==depthCounter){

break;

}

prevC = 2\*prevC;

cout << "\*\*\*\*\*\*\*\*\*\* Depth " << depthCounter << “\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n”

depthCounter++;

}

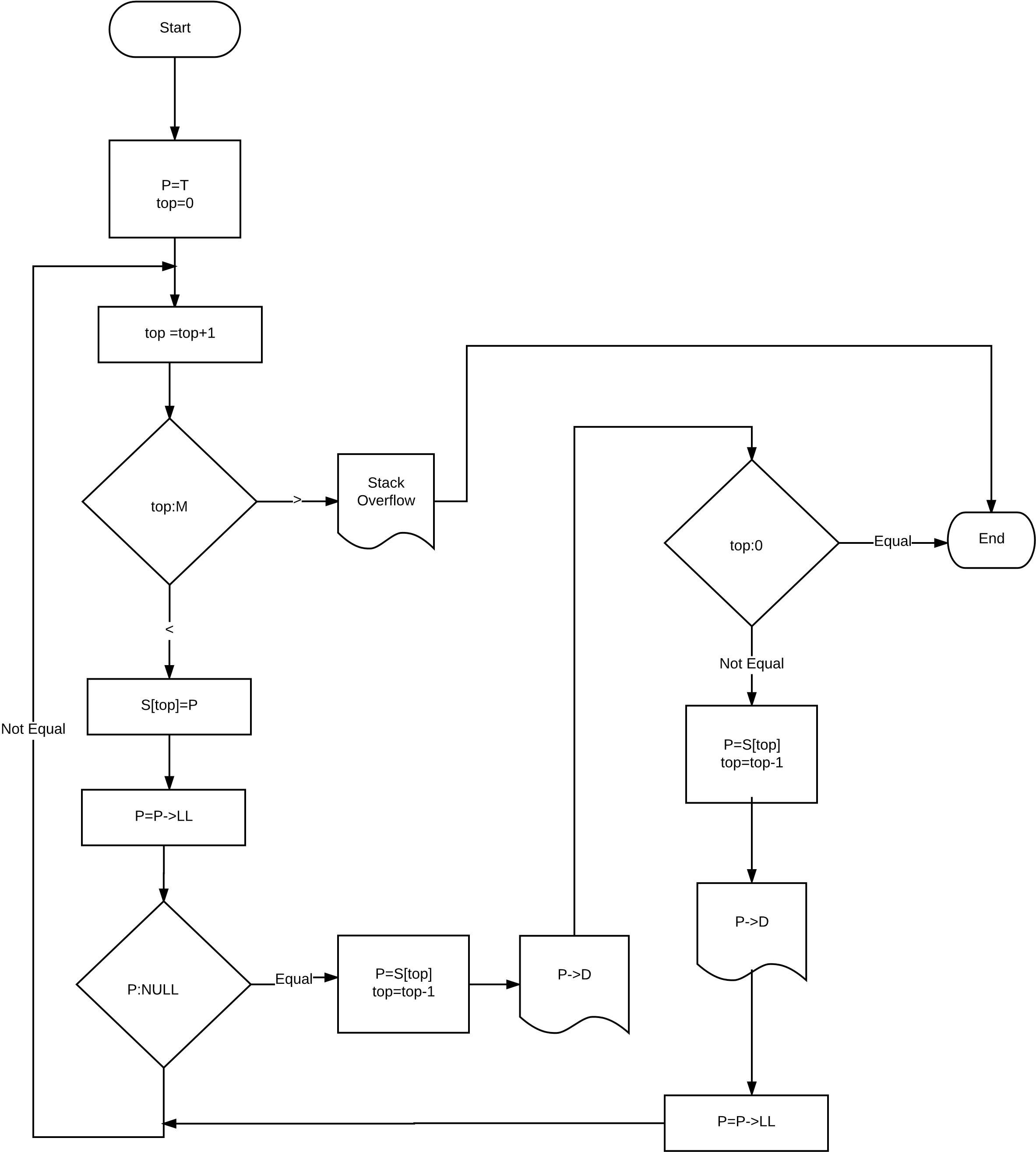
Prompt the user to enter the value of the Left Child Node. Store it in the array, index 2\*n

Prompt the user to enter the value of the Right Child Node. Store it at index (2\*n)+1

Program complete.

INORDER SEQUENCE TREE TRAVERSAL

Flowchart



CREATING A TREE WITH AN ARRAY

Source code

#include <iostream>

#include <math.h>

using namespace std;

void create\_tree(int\* n);

int main()

{

int n;

cout << "Provide Depth of Tree: \_\b";

cin >> n;

n=n+1;

cout << "\n\n";

create\_tree(&n);

return 0;

}

void create\_tree(int\* nPtr)

{

int tree\_arr[((int)pow(2,\*nPtr)-1)] = {0};

int prevC = 2;

int depthCounter = 3;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Depth 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n";

cout << "Enter value of ROOT Node: \_\b";

cin >> tree\_arr[1];

cout << "\n\*\*\*\*\*\*\*\*\*\* End of Depth \*\*\*\*\*\*\*\*\*\*\*\*\n\n\n\n";

for (int c=1;; c++){

if(c==1){

cout << "\*\*\*\*\*\*\*\*\*\*\*\* Depth 2 \*\*\*\*\*\*\*\*\*\*\*\*\n\n";

}else

{

if((2\*c)==(2\*prevC)){

if(\*nPtr==depthCounter){

break;

}

prevC = 2\*prevC;

3cout << "\*\*\*\*\*\* Depth " << depthCounter << “\*\*\*\*\*\*\*\*\n\n";

depthCounter++;

}

}

cout << "Enter value of Left Child Node of Parent [" << tree\_arr[c] << "] -> \_\b";

cin >> tree\_arr[2\*c];

cout << "Enter value of Right Child Node of Parent [" << tree\_arr[c] << "] -> \_\b";

cin >> tree\_arr[(2\*c)+1];

if(c==(prevC-1)){

cout << "\n\*\*\*\*\*\*\*\* End of Depth \*\*\*\*\*\*\*\*\n\n\n\n";

}

}

}