

Date: 10/03/2021

Practical 3:

2CSDE56 – 2CSDE56

Name: Shrey Viradiya

Roll No: 18BCE259

Aim:

Write a program to use Havel-Hakimi theorem and check whether the given degree sequence is graphical or not.

Code:

Prac3_HavelHakimi.cpp

```
// Implement the Havel Hakimi Theorm to check
// whether the given degree sequence is a graph or not

#include<iostream>
#include "HavelHakimi.h"

int main(){
    using namespace std;

    int noDegrees;
    cout << "Enter the number of degrees : ";
    cin >> noDegrees;

    cout << "Enter the degree sequence: " << endl;
    int *degrees = new int[noDegrees]{0};

    for (int i = 0; i < noDegrees; i++)
    {
        cin >> degrees[i];
    }

    applyHavelHakimi(degrees, noDegrees);

    return 0;
}
```

HavelHakimi.h

```
#pragma once

#include<iostream>
#include<algorithm>

bool applyHavelHakimi(int *degrees, int noDegrees){
    using namespace std;

    cout << "Applying Havel-Hakimi" << endl;
    cout << "-----" << endl;

    for (int i = 0; i < noDegrees; i++)
    {
        //sort the remaining
        std::sort(degrees+i, degrees+noDegrees, greater<int>());
    }
}
```

```

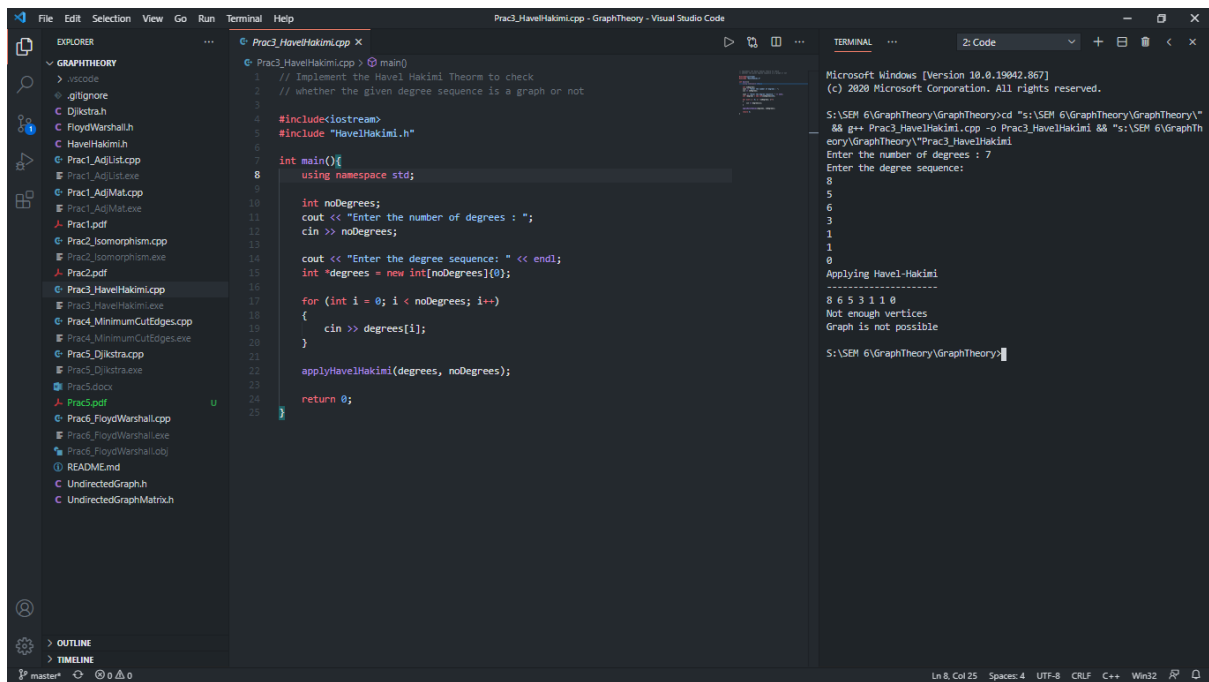
// print the remaining
for (int i = 0; i < noDegrees; i++)
{
    cout << degrees[i] << " ";
}
cout << endl;

// reduce degrees
for (int j = 0; j < degrees[i]; j++)
{
    if(i+j+1 >= noDegrees){
        cout << "Not enough vertices\nGraph is not possible" << endl;
        return false;
    }
    --degrees[i+j+1];
}

// check for ending condition
bool zeros = true;
for (int j = i+1; j < noDegrees; j++)
{
    if (degrees[j] < 0)
    {
        cout << "Negative Degree encountered\nGraph is not possible" << endl;
        return false;
    }
    if (degrees[j] != 0) zeros = false;
}
if (zeros){
    // print the remaining
    for (int i = 0; i < noDegrees; i++)
    {
        cout << degrees[i] << " ";
    }
    cout << endl;
    cout << "All the remaining are zero\nGraph is possible" << endl;
    return true;
}
}
return true;
}

```

Snapshot of the output:



```
File Edit Selection View Go Run Terminal Help
Prac3_HaveHakimi.cpp - GraphTheory - Visual Studio Code

EXPLORER
GRAPH THEORY
  .vscode
  .gitignore
  C Dijkstra.h
  C FloydWarshall.h
  C HavelHakimi.h
  C Prac1_AdjList.cpp
  F Prac1_AdjList.exe
  C Prac1_AdjMat.cpp
  F Prac1_AdjMat.exe
  Prac1.pdf
  C Prac2_Isomorphism.cpp
  F Prac2_Isomorphism.exe
  Prac2.pdf
  C Prac3_HaveHakimi.cpp
  F Prac3_HaveHakimi.exe
  C Prac4_MinimumCutEdges.cpp
  F Prac4_MinimumCutEdges.exe
  C Prac5_Dijkstra.cpp
  F Prac5_Dijkstra.exe
  Prac5.docx
  Prac5.pdf
  C Prac6_FloydWarshall.cpp
  F Prac6_FloydWarshall.exe
  Prac6_FloydWarshall.log
  README.md
  C UndirectedGraph.h
  C UndirectedGraphMatrix.h

Prac3_HaveHakimi.cpp
1 // Implement the Havel Hakimi Theorem to check
2 // whether the given degree sequence is a graph or not
3
4 #include<iostream>
5 #include "HavelHakimi.h"
6
7 int main(){
8     using namespace std;
9
10    int noDegrees;
11    cout << "Enter the number of degrees : ";
12    cin >> noDegrees;
13
14    cout << "Enter the degree sequence: " << endl;
15    int *degrees = new int[noDegrees](0);
16
17    for (int i = 0; i < noDegrees; i++)
18    {
19        cin >> degrees[i];
20    }
21
22    applyHavelHakimi(degrees, noDegrees);
23
24    return 0;
25 }
```

```
Microsoft Windows [Version 10.0.19042.867]
(c) 2020 Microsoft Corporation. All rights reserved.

S:\SEM 6\GraphTheory\GraphTheory>cd "S:\SEM 6\GraphTheory\GraphTheory\"
&& g++ Prac3_HaveHakimi.cpp -o Prac3_HaveHakimi && "S:\SEM 6\GraphTheory\GraphTheory\Prac3_HaveHakimi.exe"
Enter the number of degrees : 7
Enter the degree sequence:
8
5
6
3
1
1
0
Applying Havel-Hakimi
-----
8 6 5 3 1 1 0
Not enough vertices
Graph is not possible

S:\SEM 6\GraphTheory\GraphTheory>
```

Ln 8, Col 25 Spaces: 4 UTF-8 CRLF C++ Win32