Erdos Renyi Model

Algorithms 2 - HIT

According to what was asked, samplesNumber is set to 500, with V of 1000. And we were asked to check the following:

* Connectivity - Threshold1 - if p<Threshold1 the graph is not connected in high probability. If p>Threshold1 the graph is connected in high probability.
* Diameter - Threshold2 - if p>Threshold2 the graph has a high probability of diameter=2.

if p<Threshold2 the graph has a high probability of diameter > 2.

* Is isolated - Threshold3 - if p<Threshold3 the graph has a high probability of isolated vertex. if p>Threshold3 the graph has a high probability of non isolated vertex.

The graph was built by a Class called Graph, I created an array of sets, each set contains Edges of each Vertex.

I used pre determined probabilities to check if Edge exists or not.

Generate a random number between 0 - 1, if the number is <= to the predetermined probability, then edge exists.

Then we used different functions to check the properties of the graphs.

Explanation of each function is on the next page.

Functions explanation:

firstOption / secondOption / thirdOption

Are the three options we needed to check, we start by creating an array of P's, and array to save the result

Inside the second loop we create the relevant graph for each V(according to samplesNumber), and we check each graph inside check(first/second/third)Option function according to exercise requirements.

We check the Ratio, and export to a file.

build\_random\_graph

Build a random graph, inside the loop we use a function to check if a random probability is lower/equal to the given probability, if so we need to create an edge.

The function check\_random\_with\_prob is randoming a number from 0 <= p < 1, and returns true/false, is p <= the premade probability.

connectivity Function

Use the BFS algorithm to travel the graph, if the distance of an edge is equal to 0 = there is no edge, we return 0. If not, we return 1 = all Vertices got edges.

is\_isolated

Function checks if a graph got a Vertex without an edge, if so we return 1 to true. else is 0.

diameter

Function needs to check the longest path available between vertices and count the number of connected edges.

check(First/Second/Third)Option

Are functions to check if the terms in the exercise manual are happening or not.

oneAndThreeThreshold / secondThreshold

Are thresholds calculations according to the exercise.

exprotResultToCSV

exports result to a CSV file.

GraphBuilder.cpp

Graph::Graph() + Graph::Graph(int V)

Are constructors.

Graph::Graph

Is a copy constructor.

Graph::~Graph

Is a destructor.

Graph& Graph::operator=

Used for Operators Overloading, we define the above function as a non-member function of a class.

addEdge

Is used to add edges whenever needed. Graph is undirected - If 1 is connected to 2, 2 is connected to 1 as well.

checkVertexEdge

uses set to save values, we use set because we don't need duplicate edges. Graph is undirected.

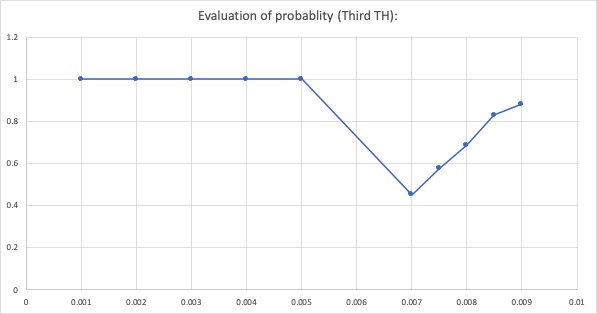
GetV

Used to return V.

BFS

BFS checks the number of edges connected, and returns an array of them.

Threshold 3 Graph:



Threshold1 Graph: