.6

```
function [time, G] = GraphTheory_Assignment2(n,T)
%Random Graph Generation process
%T=1: G -> d(v)>=1
%T=2: G -> Connectability
%T=3: G -> One Circle

%/////////////////////// Reseting
G = zeros(n);
time = 0;
a = floor(rand(1)*n)+1;
b = floor(rand(1)*n)+1;
%///////////////////////

while ((time<(n*(n-1))) && Cond(T,G))
    while ((G(a,b)==1) || (a==b))
        a = floor(rand(1)*n)+1;
        b = floor(rand(1)*n)+1;
    end
    G(a,b) = 1;                  %For Symmetry
    G(b,a) = 1;
    time = time + 1;
end
%view(biograph(G))            %//Debug
end

function [ LogicAnswer ] = Cond(T,G)
%Condotion's Validation
% LogicAnswer: 1-> False, 0-> True
%/////////////////////////////////////////////////////
if(T==1)                            %T=1: G -> d(v)>=1
    sum = 0;
    for i=1:size(G,1)
        for j=1:size(G,1)
            sum = sum + G(i,j);     %Sum all the row
        end
        if (sum==0)                 %If d(V)==0 then return false
             LogicAnswer = 1;
             return
        end
    sum = 0;
    end
    LogicAnswer = 0;   % every d(v) >= 1
end

if(T==2)                            %T=2: G -> Connectability
```

```
     S = sparse(G);
     if (graphconncomp(S,'Directed', false)==1)       %graphconncomp
returns the number of conectability elements (BFS - O(n))
          LogicAnswer = 0;
     else LogicAnswer = 1;
     end
end
end
```

7.

```
function [T1_Avarage,T2_Avarage] = Test(N,k)
%Test Performs multyply calculations on RPG Function
%N: Vector of the n's
%K: The number of repetitions for the avarage
T1_Avarage = ones(size(N,2),1)'.*0;
T2_Avarage = ones(size(N,2),1)'.*0;

for i=1:k
    for j=1:size(N,2)
        T1_Avarage(j) = T1_Avarage(j) +
GraphTheory_Assignment2(N(j),1);
        T2_Avarage(j) = T2_Avarage(j) +
GraphTheory_Assignment2(N(j),2);
    end
end
T1_Avarage = T1_Avarage/k;
T2_Avarage = T2_Avarage/k;
end
```

הרצה:

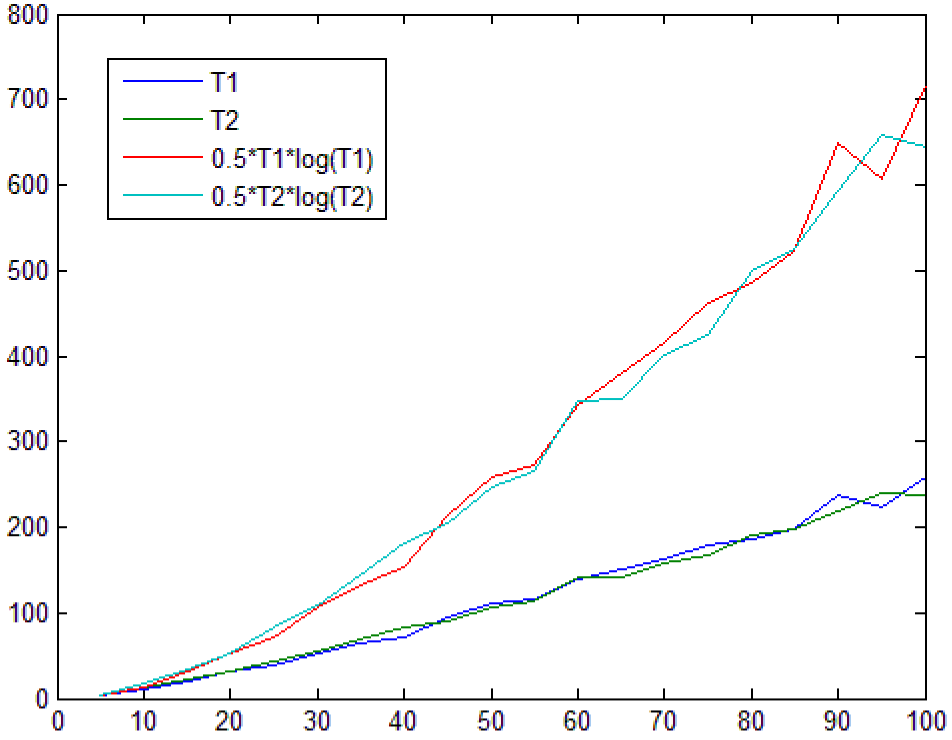
```
>> T1 = [5:5:100]

T1 =

 Columns 1 through 15

   5   10   15   20   25   30   35   40   45   50   55   60   65   70   75

 Columns 16 through 20

  80   85   90   95  100
```

```
>> [R1,R2] = Test(T1,30)

R1 =

  Columns 1 through 9

    3.8667   11.4667   20.7333   30.5333   39.0333   53.6333   63.7667   72.1000   94.5333

  Columns 10 through 18

  110.3000  115.2333  138.5333  151.0667  163.1333  178.3333  185.4333  197.4000  237.4333

  Columns 19 through 20

  224.4667  258.2333

R2 =

  Columns 1 through 9

    4.5333   12.8333   21.7000   30.5333   43.7667   54.4333   68.0667   82.0000   90.7000

  Columns 10 through 18

  105.5667  112.7667  140.6000  141.0000  158.1000  166.1000  190.2000  198.3000  219.7333

  Columns 19 through 20

  240.3333  236.2667
```

8.

הרצה:

```
>> plot(T1,R1,T1,R2,T1,P1,T1,P2)
```


800
700
600
500
400
300
200
100
0
0
10
20
30
40
50
60
70
80
90
100
T1
T2
0.5*T1*log(T1)
0.5*T2*log(T2)