Enumeration of random labeled graphs

Generated on Wed Aug 23 2023 for Enumeration of random labeled graphs by Doxygen 1.9.7

Wed Aug 23 2023 18:32:06

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 Graph_Params Struct Reference

Data Fields

- unsigned int n
- unsigned int m_min
- long double m_max
- long double m_crit
- long double n_graphs
- long double n_trees

3.1.1 Detailed Description

Graph parameters structure, declared as a new type.

Definition at line 45 of file rand_graph_lib.c.

3.1.2 Field Documentation

3.1.2.1 n

`n`

Number of labeled vertices.

Definition at line 47 of file rand_graph_lib.c.

3.1.2.2 m_min

`m_min`

Minimal number of edges in a connected graph (tree):

$m_{min} = n - 1$ (tree).

Definition at line 47 of file rand_graph_lib.c.

3.1.2.3 m_max

`m_max`

Maximal possible number of edges in the complete graph with n labeled vertices:

$m_{max} = \binom{n}{2} = \frac{n(n-1)}{2}$ (complete graph).

Definition at line 48 of file rand_graph_lib.c.

3.1.2.4 m_crit

`m_crit`

Number of edges as connectedness threshold:

$m_{crit} = \binom{n-1}{2} = \frac{(n-1)(n-2)}{2}$.

Definition at line 48 of file rand_graph_lib.c.

3.1.2.5 n_graphs

`n_graphs`

Maximal possible number of graphs with n labeled vertices:

$n_{graphs} = 2^{m_{max}}$.

Definition at line 48 of file rand_graph_lib.c.

3.1.2.6 n_trees

`n_trees`

Maximal possible number of trees with n labeled vertices [3] :

$n_{trees} = n^{n-2}$.

Definition at line 48 of file rand_graph_lib.c.

The documentation for this struct was generated from the following file:

- rand_graph_lib.c

Chapter 4

File Documentation

4.1 rand_graph_lib.c File Reference

```
#include <math.h>
```

Data Structures

- struct Graph_Params

Typedefs

- typedef struct Graph_Params GP

Functions

- GP calc_graph_params (unsigned int n)
- long double A006125_total (unsigned int n)
- long double A001187_conn (unsigned int n)
- long double A054592_disconn (unsigned int n)
- float prob_conn (unsigned int n, float p)

4.1.1 Detailed Description

Enumerate and calculate the probability of connectedness of random graphs constructed with the Erdős–Rényi and Gilbert models.

Author

Andrei Batyrov (arbatyrov@edu.hse.ru)

Version

0.1

Date

2023-08-23

Copyright

Copyright (c) 2023

Definition in file rand_graph_lib.c.

4.1.2 Typedef Documentation

4.1.2.1 GP

```
typedef struct Graph_Params GP
```

4.1.3 Function Documentation

4.1.3.1 calc_graph_params()

```
GP calc_graph_params (
            unsigned int n )
```

Populate and return graph parameters as a structure.

Parameters

n	Number of labeled vertices

Returns

Graph parameters as per structure Graph_Params

Definition at line 57 of file rand_graph_lib.c.

4.1.3.2 A006125_total()

```
long double A006125_total (
            unsigned int n )
```

Find the total number of labeled graphs (connected and disconnected) with n nodes – sequence $A006125$ [4] .

Parameters

n	Graph order – number of vertices

Returns

Number of labeled graphs

Implementation notes:

1. the number of labeled graphs (connected and disconnected) with n nodes is GP::n_graphs.
2. Observe, that $\binom{n}{2} = \frac{n(n-1)}{2}$, thus no factorial is used here.
3. Time complexity: $\mathcal{O}(\text{pow})$.

Definition at line 79 of file rand_graph_lib.c.

4.1.3.3 A001187_conn()

```
long double A001187_conn (
            unsigned int n )
```

Find the number of connected labeled graphs with n nodes constructed with the Erdős–Rényi model $\mathcal{G}(n, M)$ – sequence $A001187$ [5] .

Parameters

n	Graph order – number of vertices

Returns

Number of connected labeled graphs

Implementation notes:

1. In this model each graph is chosen randomly with equal probability of $1/n_{graphs}$, where n_{graphs} is GP::n_graphs.
2. The number C_n of labeled connected graphs of order n is given by the recursive formula (1.2.1), p. 7 [1] ; p was substituted with n to avoid confusion with notation of probability.
   $$C_n = 2^{\binom{n}{2}} - \frac{1}{n}\sum_{k=1}^{n-1} k\binom{n}{k}2^{\binom{n-k}{2}}C_k.$$
3. Observe, that $n! = \Gamma(n+1)$, thus the standard C `tgammal()` function is used.
4. Observe, that $\binom{n}{2} = \frac{n(n-1)}{2}$, thus no factorial is used here.
5. Time complexity: $\mathcal{O}(2^{n\mathcal{O}(\text{tgamma + pow})-1})$.

Definition at line 102 of file rand_graph_lib.c.

4.1.3.4 A054592_disconn()

```
long double A054592_disconn (
            unsigned int n )
```

Find the number of disconnected labeled graphs with n nodes – sequence $A054592$ [6] .

Parameters

n	Graph order – number of vertices

Returns

Number of disconnected labeled graphs

Implementation notes:

1. Number of labeled graphs (connected and disconnected) with n nodes is sequence $A006125$ [4] . Number of connected labeled graphs with n nodes is sequence $A001187$ [5] . Thus, the number of disconnected labeled graphs with n nodes is simply $A054592(n) = A006125(n) - A001187(n)$.
2. Time complexity: $\max(\mathcal{O}(\text{ A006125\_total }), \mathcal{O}(\text{ A001187\_conn }))$.

Definition at line 131 of file rand_graph_lib.c.

4.1.3.5 prob_conn()

```
float prob_conn (
            unsigned int n,
            float p )
```

Find the probability of connectedness of a random labeled graph constructed with the Gilbert model $\mathcal{G}(n, p)$.

Parameters

n	Graph order – number of vertices
p	Edge probability $0 \leq p \leq 1$

Returns

$P_N = P(\mathcal{G}(n, p)$ is connected$)$

Implementation notes:

1. In this model every possible edge occurs independently with probability p. The probability of obtaining any one particular random graph with m edges is $p^m(1-p)^{N-m}$, where N is GP::m_max.
2. The probability of connectedness of a random labeled graph is given by the recursive formula (3) on p. 2 [2] ; q was substituted by $1-p$ to avoid introducing unnecessary new variable.
$$1 - P_N = \sum_{k=1}^{N-1} \binom{N-1}{k-1} P_k q^{k(N-k)}.$$
3. Observe, that $n! = \Gamma(n+1)$, thus the standard C `tgammal()` function is used.
4. Time complexity: $\mathcal{O}(2^{n\mathcal{O}(\text{tgamma + pow})-1})$.

Definition at line 155 of file rand_graph_lib.c.

4.2 rand_graph_lib.c

Go to the documentation of this file.

```
00001
00011 #include <math.h>
00012
00045 typedef struct Graph_Params
00046 {
00047     unsigned int n, m_min;
00048     long double m_max, m_crit, n_graphs, n_trees;
00049 } GP;
00050
```

```
00057 GP calc_graph_params(unsigned int n)
00058 {
00059     GP gp;
00060     gp.m_max = n * (n - 1) / 2;
00061     gp.m_min = n - 1;
00062     gp.m_crit = (n - 1) * (n - 2) / 2;
00063     gp.n_graphs = pow(2, gp.m_max);
00064     gp.n_trees = pow(n, (n - 2));
00065     return gp;
00066 }
00067
00079 long double  A006125_total(unsigned int n)
00080 {
00081     return pow(2, (n * (n - 1) / 2));
00082 }
00083
00102 long double A001187_conn(unsigned int n)
00103 {
00104     unsigned int k;
00105     long double disconn_count = 0.0;
00106
00107     if (n == 1)
00108     {
00109         return 1.0;
00110     }
00111     for (k = 1; k < n; k++)
00112     {
00113         disconn_count += k * (tgammal(n + 1) / tgammal(k + 1) / tgammal(n - k + 1)) * pow(2, ((n - k)
      * (n - k - 1) / 2)) * A001187_conn(k);
00114     }
00115     return pow(2, (n * (n - 1) / 2)) - disconn_count / n;
00116 }
00117
00131 long double A054592_disconn(unsigned int n)
00132 {
00133     return A006125_total(n) - A001187_conn(n);
00134 }
00135
00155 float prob_conn(unsigned int n, float p)
00156 {
00157     // float q = 1.0 - p;
00158     float prob_disconn = 0.0;
00159     unsigned int k;
00160     for (k = 1; k < n; k++)
00161     {
00162         prob_disconn += tgammal(n) / tgammal(k) / tgammal(n - k + 1) * pow((1 - p), (k * (n - k))) *
      prob_conn(k, p);
00163     }
00164     return 1.0 - prob_disconn;
00165 }
```

Bibliography

[1] Frank Harary, Edgar M. Palmer. *Graphical Enumeration*. Academic Press, 1973. 9

[2] E. N. Gilbert. *Random Graphs*. Bell Telephone Laboratories, Inc., 1959. 10

[3] P. Erdős, A. Rényi. *On Random Graphs, p. 292*. Budapest, 1959. 6

[4] N. J. A. Sloane. *a(n) = 2 ^n*(n-1)/2) (Formerly M1897)*. The On-Line Encyclopedia of Integer Sequences, 1991. 8, 10

[5] N. J. A. Sloane. *Number of connected labeled graphs with n nodes*. The On-Line Encyclopedia of Integer Sequences, 1991. 9, 10

[6] N. J. A. Sloane. *Number of disconnected labeled graphs with n nodes*. The On-Line Encyclopedia of Integer Sequences, 2000. 9

Index