Enumeration of random labeled graphs

Generated on Fri Aug 25 2023 for Enumeration of random labeled graphs by Doxygen 1.9.7

Fri Aug 25 2023 23:21:44

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 short n
- unsigned short m_min
- unsigned short m_max
- unsigned short m_crit
- unsigned long long n_graphs
- unsigned long long n_trees

3.1.1 Detailed Description

Graph parameters structure, declared as a new type.

Definition at line 88 of file rand_graph_lib.c.

3.1.2 Field Documentation

3.1.2.1 n

`n`

Number of labeled vertices.

Definition at line 90 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 90 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 90 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 90 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 91 of file rand_graph_lib.c.

3.1.2.6 n_trees

n_trees

Maximal possible number of trees with n labeled vertices, p.292, Erdős [3] :

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

Definition at line 91 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

- unsigned long long binom (unsigned short n, unsigned short k)
- GP calc_graph_params (unsigned short n)
- unsigned long long A006125_total (unsigned short n)
- unsigned long long A001187_conn (unsigned short n)
- unsigned long long A054592_disconn (unsigned short n)
- double prob_conn (unsigned short n, double 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

2023-08-25

Copyright

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 binom()

```
unsigned long long binom (
            unsigned short n,
            unsigned short k )
```

Find binomial coefficient $C(n, k)$.

Parameters

n	Integer number n
k	Integer number k

Returns

Binomial coefficient

Implementation notes:

1. Optimized algorithm without explicit calculation of factorial.
2. Integer overflow unsafe, since the result size is not checked during calculation and before return.
3. Time complexity: $\mathcal{O}(r)$, where $r = \min(k, n - k)$.

Definition at line 24 of file rand_graph_lib.c.

4.1.3.2 calc_graph_params()

```
GP calc_graph_params (
            unsigned short n )
```

Populate and return graph parameters as a structure.

Parameters

n	Number of labeled vertices

Returns

Graph parameters as per structure Graph_Params

Implementation notes:

1. Integer overflow unsafe, since the rhs's size is not checked before assignment.

Definition at line 103 of file rand_graph_lib.c.

4.1.3.3 A006125_total()

```
unsigned long long A006125_total (
            unsigned short 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. Integer overflow unsafe, since the result's size is not checked before return.
3. Time complexity: $\mathcal{O}(\text{pow})$.

Results for $n \in [0, 11]$:

n	A006125(n)
0	1
1	1
2	2
3	8
4	64
5	1 024
6	32 768
7	2 097 152
8	268 435 456
9	68 719 476 736
10	35 184 372 088 832
11	36 028 797 018 963 968

Definition at line 141 of file rand_graph_lib.c.

4.1.3.4 A001187_conn()

```
unsigned long long A001187_conn (
            unsigned short 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, Harary [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. Integer overflow unsafe, since the result's size is not checked during calculation and before return.
4. Time complexity: $\mathcal{O}(2^{n \max(\mathcal{O}(\text{binom}),\mathcal{O}(\text{pow}))-1})$.

Results for $n \in [0, 11]$:

n	A001187(n)
0	1
1	1
2	1
3	4
4	38
5	728
6	26 704
7	1 866 256
8	251 548 592
9	66 296 291 072
10	34 496 488 594 816
11	35 641 657 548 953 344

Definition at line 183 of file rand_graph_lib.c.

4.1.3.5 A054592_disconn()

```
unsigned long long A054592_disconn (
            unsigned short 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. Integer overflow unsafe, since the result's size is not checked during calculation and before return.
3. Time complexity: $\max(\mathcal{O}($ A006125_total $), \mathcal{O}($ A001187_conn $))$.

Results for $n \in [0, 11]$:

n	A054592(n)
0	0
1	0
2	1
3	4
4	26
5	296
6	6 064
7	230 896
8	16 886 864
9	2 423 185 664
10	687 883 494 016
11	387 139 470 010 624

Definition at line 229 of file rand_graph_lib.c.

4.1.3.6 prob_conn()

```
double prob_conn (
            unsigned short n,
            double 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), p. 2, Gilbert [2] ; q was substituted by $1-p$ to avoid introducing unnecessary new variable; N was substituted by n to avoid confusion with N above.
$$P_n = 1 - \sum_{k=1}^{n-1} \binom{n-1}{k-1} (1-p)^{k(n-k)} P_k.$$
3. Integer overflow unsafe, since the result's size is not checked during calculation and before return.
4. Time complexity: $\mathcal{O}(2^{n\max(\mathcal{O}(\text{binom}),\mathcal{O}(\text{pow}))-1})$.

Results for P_n for $n \in [2,11]$ and $p \in [0.1,0.9]$:

n / p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
2	0.10000	0.20000	0.30000	0.40000	0.50000	0.60000	0.70000	0.80000	0.90000
3	0.02800	0.10400	0.21600	0.35200	0.50000	0.64800	0.78400	0.89600	0.97200
4	0.01293	0.08250	0.21865	0.40038	0.59375	0.76550	0.89249	0.96666	0.99581
5	0.00810	0.08195	0.25626	0.48965	0.71094	0.87026	0.95751	0.99166	0.99949
6	0.00621 †	0.09230	0.31690	0.59555	0.81494 ‡	0.93652	0.98497	0.99805	0.99994
7	0.00551	0.11127	0.39385	0.69878	0.88990	0.97072	0.99484	0.99955	0.99999
8	0.00541	0.13851	0.47987	0.78627	0.93709	0.98677	0.99824	0.99990	1.00000
9	0.00574	0.17396	0.56714	0.85325	0.96474	0.99408	0.99941	0.99998	1.00000
10	0.00644	0.21723	0.64897	0.90128	0.98045	0.99737	0.99980	0.99999	1.00000
11	0.00752	0.26729	0.72107	0.93445	0.98925	0.99885	0.99994	1.00000	1.00000

† Table 1, p. 2, Gilbert [2] : 0.00624.

‡ Table 1, p. 2, Gilbert [2] : 0.81569.

Definition at line 276 of file rand_graph_lib.c.

4.2 rand_graph_lib.c

Go to the documentation of this file.

```
#include <math.h>

unsigned long long binom(unsigned short n, unsigned short k)
{
    unsigned short i;
    unsigned long long coeff = 1;

    /* Special cases */
    if (k == 0 | k == n)
    {
        return coeff;
    }
    if (k == 1 | k == n - 1)
    {
```

```c
        return n;
    }
    if (k == 2 | k == n - 2)
    {
        return n * (n - 1) / 2;
    }

    /* General case */
    if (k > n - k)
    {
        k = n - k;
    }
    for (i = 0; i < k; i++)
    {
        coeff *= (n - i);
        coeff /= (i + 1);
    }
    return coeff;
}

typedef struct Graph_Params
{
    unsigned short n, m_min, m_max, m_crit;
    unsigned long long n_graphs, n_trees;
} GP;

GP calc_graph_params(unsigned short n)
{
    GP gp;
    gp.m_max = binom(n, 2);
    gp.m_min = n - 1;
    gp.m_crit = binom(n - 1, 2);
    gp.n_graphs = pow(2, gp.m_max);
    gp.n_trees = pow(n, n - 2);
    return gp;
}

unsigned long long A006125_total(unsigned short n)
{
    if (n == 0 | n == 1)
    {
        return 1;
    }
    return pow(2, binom(n, 2));
}

unsigned long long A001187_conn(unsigned short n)
{
    unsigned short k;
    unsigned long long disconn_count = 0;

    if (n == 0 | n == 1 | n == 2)
    {
        return 1;
    }
    for (k = 1; k < n; k++)
    {
        disconn_count += k * binom(n, k) * pow(2, binom(n - k, 2)) * A001187_conn(k);
    }
    return pow(2, binom(n, 2)) - disconn_count / n;
}

unsigned long long A054592_disconn(unsigned short n)
{
    if (n == 0 | n == 1)
    {
        return 0;
    }
    return A006125_total(n) - A001187_conn(n);
}

double prob_conn(unsigned short n, double p)
{
    // double q = 1.0 - p;
    double prob_disconn = 0.0;
    unsigned short k;
    for (k = 1; k < n; k++)
    {
        prob_disconn += binom(n - 1, k - 1) * pow(1 - p, k * (n - k)) * prob_conn(k, p);
    }
    return 1.0 - prob_disconn;
}
```

Bibliography

[1] Frank Harary, Edgar M. Palmer. *Graphical Enumeration*. Academic Press, 1973. 10

[2] E. N. Gilbert. *Random Graphs*. Bell Telephone Laboratories, Inc., 1959. 12

[3] P. Erdős, A. Rényi. *On Random Graphs*. 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. 9, 11

[5] N. J. A. Sloane. *Number of connected labeled graphs with n nodes*. The On-Line Encyclopedia of Integer Sequences, 1991. 10, 11

[6] N. J. A. Sloane. *Number of disconnected labeled graphs with n nodes*. The On-Line Encyclopedia of Integer Sequences, 2000. 11

Index