Grand graph challenge

Consider an **Erdős–Rényi random graph** with n nodes and $\frac{n \ln n}{8}$ links. Implement an **efficient algorithm** that constructs such a graph and computes the fraction of nodes in the giant connected component S. The algorithm can use any graph representation and any method for computing connected components. It should merely **compute S for a given n**. What is the size of the **largest graph** you are able to analyze in roughly **one minute**?

Nodes n	**Links $\frac{n \ln n}{8}$**	**Giant S**	**Graph T_G**	**Search T_S**	**Time T_{G+S}**
1000	863	70.32%	0.00 sec	0.00 sec	0.00 sec
10000	11 513	86.46%	0.00 sec	0.00 sec	0.01 sec
100 000	143 912	93.14%	0.03 sec	0.03 sec	0.06 sec
1 000 000	1 726 939	96.42%	0.63 sec	0.53 sec	1.16 sec
5 000 000	9 640 593	97.69%	10.18 sec	3.70 sec	13.88 sec
10 000 000	20 147 620	98.08%	15.19 sec	10.43 sec	25.62 sec
15 000 000	30 981 676	98.28%	21.79 sec	13.59 sec	35.38 sec
20 000 000	**42 028 107**	**98.40%**	**31.37 sec**	**19.94 sec**	**51.31 sec**
25 000 000	53 232 457	98.49%	42.00 sec	30.32 sec	72.32 sec
30 000 000	64 562 655	98.56%	66.23 sec	44.30 sec	110.53 sec
50 000 000	110 797 085	98.74%	232.47 sec	113.79 sec	346.26 sec

Table shows performance of a Java implementation run on 2.3 GHz Intel Core i7 with 16 GB memory.