Giantbook - Report

February 15, 2016

Report Group AB:
Adrian Brink,
Asger Pedersen,
Simon Flachs,
Troels Møller

Results

The following table summarises our results. It shows the average number of random connections needed before the emergence of the giant component ("giant"), the disappearance of the last isolated individual ("no isolated"), and when the network becomes connected ("connect").

N	T	giant	(stddev)	non-isolated	(stddev)	connected	(stddev)
100	100	7.15×10^1	5.9	5.88×10^2	6.64×10^1	2.69×10^2	6.40×10^1
1000	100	6.94×10^2	1.66×10^1	3.82×10^3	6.95×10^2	3.83×10^3	5.97×10^2
10000	100	6.92×10^3	5.75×10^1	4.87×10^4	6.22×10^3	4.87×10^4	6.22×10^3
100000	100	6.93×10^4	1.56×10^2	6.15×10^5	6.62×10^4	6.15×10^5	6.62×10^4
1000000	10	6.93×10^5	7.01×10^2	6.89×10^6	4.88×10^5	6.89×10^6	4.88×10^5
10000000	10	6.93×10^6	1.72×10^3	8.39×10^7	4.81×10^6	8.39×10^7	4.81×10^6

Our main findings are the following: The first thing that happens is that the giant component emerges, which done by regression looks like:

$$y = 8{,}420414x - 334445{,}4$$

Perhaps surprisingly, two of the events seem to happen simultaneously, namely non-isolated and connected, which happen at time linear to N.

Implementation details

We have based our union-find data type on WeightedQuickUnionUF.`java` from Sedgewick and Wayne: *Algorihthms, 4th ed.*, Addison-Wesley (2011). We added two attributes to MyUnionFind:

- a maxCompenentSize()
- a set called isolated(), which contains all the isolated sites

In the constructor the isolated set is populated with all the sites, since at initialization they are all isolated. Also The max component size is

set to 1, since at initialization none of the sites are connected. in the union method, we update the max component size if the newly created component is larger than the current max component size. Also both of the sites that have been united are now removed from the isolated set, since now neither of these are isolated.
We also added two methods: maxComponentSize which returns the value of the attribute of the same name and isNonIsolated, which returns if the set of isolated sites is empty.

Assuming we never run out of memory or heap space, if we would let our algorithm for detecting the emergence of a giant component run for 24 hours, it could compute the answer for N = 46,685,500,000

To calculate this we modified the algorithm to break when the giant component emerges, and measured it again.

N:	Time (sec.)
100	0.261
1000	0.146
10000	0.161
100000	0.220
1000000	1.023
10000000	18.5

Linear regression: y = 540340.8 x + 22706.49

y = 540340.8 * 86400 + 22706.49 = 46,685,500,000

Discussion

We defined the giant component to have size at least αN for $\alpha = \frac{1}{2}$. The choice of constant is important; choosing 0.01 and 0.99 changes the experiment completely because the average number of operations for giant component to emerge change from 49005.36 to 232284.48 for N = 100000 and T = 100.