Mogussort

😤😩😩💦💦💦

1. 📮Mogussort

World's (📮)SUSSIEST sorting algorithm

2. 📮Lore📮

Your normal and boring sorting algorithms are just too fast? You want to introduce some sussy code to your programs?
Then you have come to the right place.

Inspired by the fantastic Csussus version and certified 📮 extra sussy📮 by the original author!

Hi! I'm Sussy Baka!
photo/drawing of you/your oc/fursona
These are my Pronouns: Sus
And this is my Sexuality:
Color the flag(s)
(optional)
Here are some of my favorite things: Not being sus
Red being sus
Doing tasks
Crew winning
And these are some of the things I like to do: Win
Try to not be sus
Make red sus
This is me! And I'm proud of who I am!
imgflip.com

sussy

3. 📮How it works📮

There are 2 (two, 2📮) lists, `crew_in` and `crew_out`.

- `crew_in` contains all crewmates at the start of the game.
- `crew_out` is empty.

The algorithm will vote out one random crewmate and put it in `crew_out` That crewmate is then removed from `crew_in`. Then, it checks if `crew_out` is correctly sorted (ascending).

If it is correnct, it will continue to vote and populate `crew_out`.

If the crewmates are not in the right order an 📮Impostor📮 (sus, baka) has been spotted. This means that the game is lost, and `crew_out` will be cleared, and `crew_in` will be reset to the original form.

The algorithm will stop once all crewmates are in the right order.

4. 🚪Performance🚪

The 🚪mogussort solves all of your sorting problems with an incredible, luck based runtime. You might ask "how fast is it exactly?".

Well, having c crew mates the probability p of ~~sorting~~ voting correctly in 1 ~~attempts to sort~~ emergency meeting is:

$$\begin{aligned} p &= \frac{1}{c} \cdot \frac{1}{c-1} \cdot \frac{1}{c-2} \cdot \ldots \cdot \frac{1}{1} \tag{1} \\ &= \frac{1}{c!} \tag{2} \end{aligned}$$

That means the probability to fail in 1 emergency meeting is:

$$1 - p = 1 - \frac{1}{c!} \tag{3}$$

If we make n emergency meetings, the chance P of 1 success is:

$$\begin{aligned} P &= 1 - (1-p)^n \tag{4} \\ &= 1 - (1 - \frac{1}{c!})^n \tag{5} \end{aligned}$$

So, how many meetings do we need until we have a chance of P to have them sorted correctly once?

$$\begin{aligned} P &= 1 - (1 - \frac{1}{c!})^n \tag{6} \\ (1 - \frac{1}{c!})^n &= 1 - P \tag{7} \\ n &= \log_{1-\frac{1}{c!}}(1-P) \tag{8} \\ n &= \frac{\lg(1-P)}{\lg(1-\frac{1}{c!})} \tag{9} \end{aligned}$$

So if we manage m meetings per second, mogussort will finish with a probability P within time t_P :

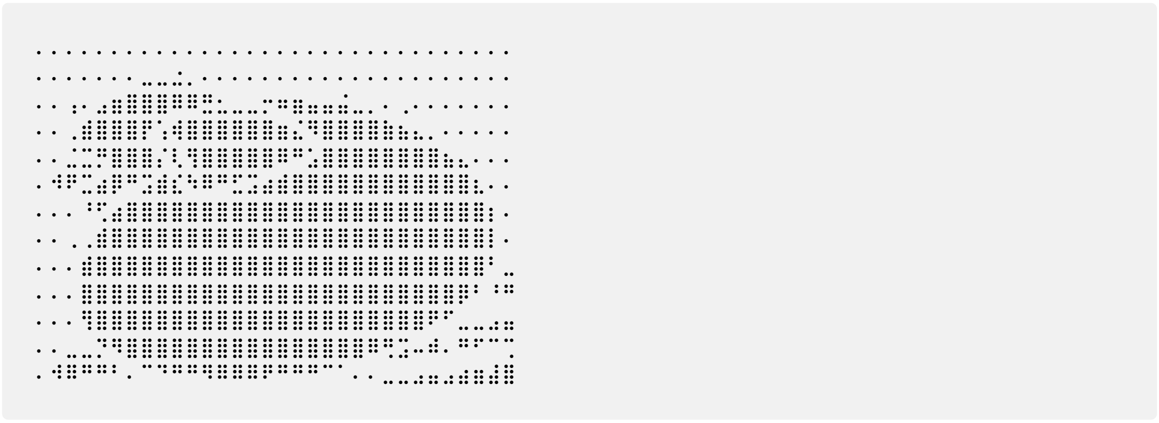
$$\begin{aligned} t_P &= \frac{n}{m} \tag{10} \\ &= \frac{\lg(1-P)}{\lg(1-\frac{1}{c!})} \cdot \frac{1}{m} \tag{11} \end{aligned}$$

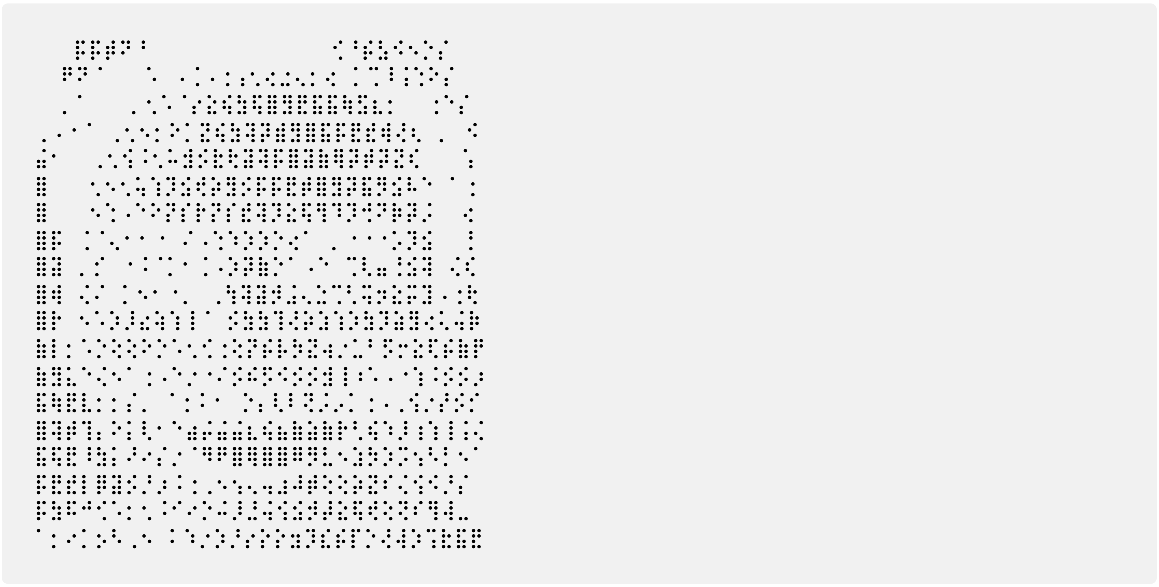
Let's assume we manage $m = 60\mathrm{k}\frac{\mathrm{meeting}}{\mathrm{s}}$, which is a value I roughly get on my computer (Intel Core i7-8550U). Then mogussort will finish with a probability P of $0,50$ and $0,95$ within:

Crew Mates c	**Time** $t_{0,50}$	**Time** $t_{0,50}$	**Time** $t_{0,95}$	**Time** $t_{0,95}$
2	$16,67\mu\mathrm{s}$		$72,03\mu\mathrm{s}$	
3	$63,36\mu\mathrm{s}$		$273,9\mu\mathrm{s}$	
4	$271,4\mu\mathrm{s}$		$1,173\mathrm{ms}$	
5	$1,381\mathrm{ms}$		$5,966\mathrm{ms}$	
6	$8,312\mathrm{ms}$		$35,92\mathrm{ms}$	
7	$58,22\mathrm{ms}$		$251,6\mathrm{ms}$	
8	$465,8\mathrm{ms}$		$2,013\mathrm{s}$	
9	$4,192\mathrm{s}$		$18,12\mathrm{s}$	
10	$41,92\mathrm{s}$		$181,2\mathrm{s}$	$(3,0\mathrm{min})$
11	$461,1\mathrm{s}$	$(7,7\mathrm{min})$	$1,993\mathrm{ks}$	$(33\mathrm{min})$
12	$5,534\mathrm{ks}$	$(1,5\mathrm{h})$	$23,92\mathrm{ks}$	$(6,6\mathrm{h})$
13	$71,94\mathrm{ks}$	$(20\mathrm{h})$	$310,9\mathrm{ks}$	$(3,6\mathrm{d})$
14	$1,007\mathrm{Ms}$	$(12\mathrm{d})$	$4,353\mathrm{Ms}$	$(50\mathrm{d})$
15	$15,11\mathrm{Ms}$	$(170\mathrm{d})$	$65,29\mathrm{Ms}$	$(2,1\mathrm{a})$
16	$242,0\mathrm{Ms}$	$(7,7\mathrm{a})$	$1,046\mathrm{Gs}$	$(33\mathrm{a})$
17	$4,162\mathrm{Gs}$	$(132\mathrm{a})$	$17,99\mathrm{Gs}$	$(570\mathrm{a})$
18	$104,1\mathrm{Gs}$	$(3.297\mathrm{a})$	$449,7\mathrm{Gs}$	$(14.251\mathrm{a})$

This even means you might be able to play a game of Among Us while you wait for the algorithm to finish (now isn't that fantastic?).

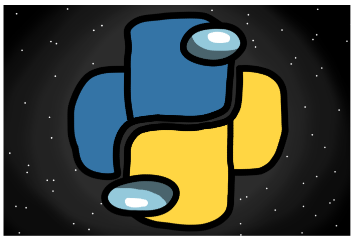
5. Sus





6. Language

Made in py sus to improve performance in comparison to the original Csussus version.



pysus