

STRANGE ALGORITHMS COMPARISON



ADA FINAL PROJECT

Final Presentation

oooo
Jocelin | Nicholas | Tiffany

TABLE OF CONTENTS

- Background
- Problems
- Proposed Algorithms
- Process
- Results
- Comparison
- Conclusion



BACKGROUND

A sorting algorithm is an algorithm made up of a series of instructions that takes an array as input, performs specified operations on the array, and outputs a sorted array.

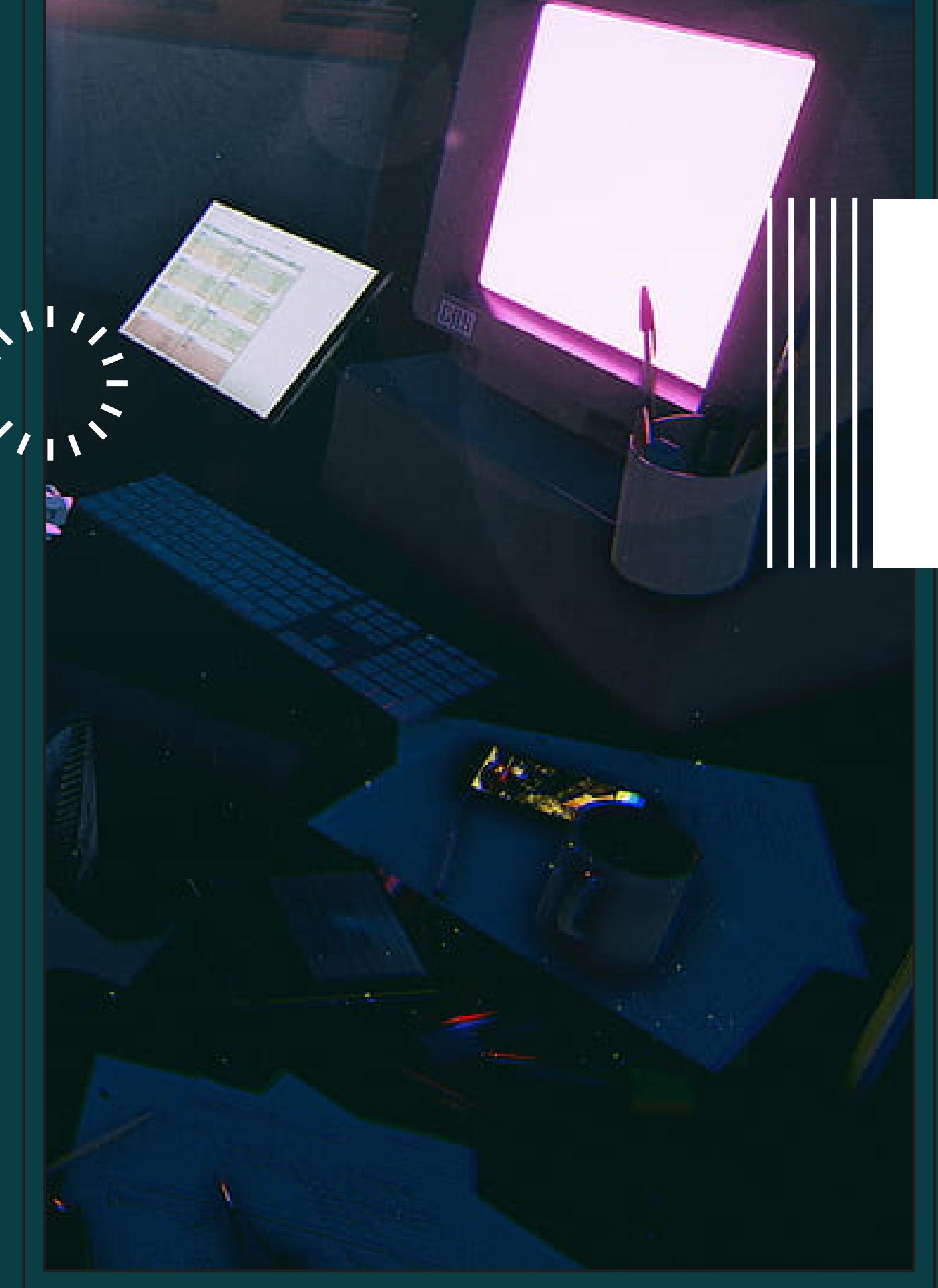


STRANGE ALGORITHMS

How do we select them?

- Not often heard
- Algorithms created for jokes
- Unusual names





Problem

Find out whether strange algorithms
can be used practically and if not why.

Find which algorithm have the fastest
run time and the most memory efficient
+ vice versa.

PROPOSED ALGORITHMS



STALIN

Eliminating numbers
that are not positioned
correctly.



SLEEP

The element having the least
amount of sleeping time wakes
up first and the number gets
printed.



COCKTAIL SHAKER

A variation of bubble sort. It
traverses through a given array in
both directions.

PROCESS

PROCESS 01

Prepare the code algorithm and various arrays (nearly sorted and random)

PROCESS 02

Measure the run time and physical memory usage.

PROCESS 03

Record the run time and physical memory with tables and test the arrays 5 times, score out the average.

PROCESS 04

Comparison between memory, time, code simplicity, and time complexity

PROCESS 05

Identifying pros and cons of each algorithm

PROCESS 06

We conclude whether the algorithms can be applied or not and in which situation is the algorithms best at

STALIN RESULTS

STALIN SORT (Partially)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY	PEAK MEMORY
1	1000	357	4202496	4206592
2	1000	377	4190208	4194304
3	1000	351	4202496	4206592
4	1000	361	4188112	4190208
5	1000	329	4188112	4190208
Average:		355	4193484.8	4197580.8
1	10000	28426	4390912	4395008
2	10000	26701	4382720	4386816
3	10000	26620	4382720	4386816
4	10000	30304	4337664	4382720
5	10000	26898	4395008	4384550
Average:		27389.8	4377804.8	4387182
1	15000	96690	6230016	6234112
2	15000	155114	4444160	4448256
3	15000	93319	4448256	4452352
4	15000	93957	4448256	4452352
5	15000	143729	4457213	4469237
Average:		116561.8	4805580.2	4811261.8

STALIN SORT(Random)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY	PEAK MEMORY
1	1000	597	4206592	4210688
2	1000	607	4202496	4206592
3	1000	587	4202496	4206592
4	1000	591	4182016	4186112
5	1000	542	4194304	4198400
Average:		584.8	4197580.8	4201676.8
1	10000	154274	4452352	4456448
2	10000	163551	4460544	4464640
3	10000	155469	4452352	4456448
4	10000	145679	4357098	4464895
5	10000	68066	4329472	4333568
Average:		137407.8	4410363.6	4435199.8
1	15000	154474	4456448	4460544
2	15000	593149	4481024	4485120
3	15000	163803	4460544	4464640
4	15000	180105	4448256	4452352
5	15000	1913791	4063232	4067328
Average:		601064.4	4381900.8	4385996.8

COCKTAIL RESULTS

COCKTAIL SORT (Partially)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY	PEAK MEMORY
1	1000	28	3235840	3235840
2	1000	33	3231744	3231744
3	1000	4	3235840	3235840
4	1000	28	3235840	3235840
5	1000	37	3231744	3231744
Average:		26	3234201.6	3234201.6
1	10000	508	3301376	3301376
2	10000	438	3309568	3309568
3	10000	448	3297280	3297280
4	10000	660	3305472	3305472
5	10000	672	3305472	3305472
Average:		545.2	3303833.6	3303833.6
1	15000	817	3346432	3346432
2	15000	913	3342336	3342336
3	15000	942	3338240	3338240
4	15000	770	3342336	3342336
5	15000	890	3346432	3346432
Average:		866.4	3343155.2	3343155.2
1	30000	1880	3469312	3469312
2	30000	1986	3465216	3465216
3	30000	1861	3469312	3469312
4	30000	1422	3461120	3461120
5	30000	2127	3465216	3465216
Average:		1855.2	3466035.2	3466035.2
1	50000	4589	3624960	3624960
2	50000	5138	3624960	3624960
3	50000	4498	3624960	3624960
4	50000	5469	3624960	3624960
5	50000	5420	3629056	3629056
Average:		5022.8	3625779.2	3625779.2

COCKTAIL SORT (Random)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY	PEAK MEMORY
1	1000	5954	3235840	3235840
2	1000	6057	3235840	3235840
3	1000	5208	3235840	3235840
4	1000	5765	3235840	3235840
5	1000	4536	3235840	3235840
Average:		5504	3235840	3235840
1	10000	450244	3297280	3297280
2	10000	466558	3305472	3305472
3	10000	480377	3305472	3305472
4	10000	422027	3305472	3305472
5	10000	456552	3305472	3305472
Average:		455151.6	3303833.6	3303833.6
1	15000	1041002	3338240	3338240
2	15000	1015040	3342336	3342336
3	15000	1006960	3342336	3342336
4	15000	1025891	3346432	3346432
5	15000	1000227	3342336	3342336
Average:		1017824	3342336	3342336
1	30000	4501050	3461120	3461120
2	30000	3790061	3465216	3465216
3	30000	3829977	3469312	3469312
4	30000	3803590	3461120	3461120
5	30000	3828456	3469312	3469312
Average:		3950626.8	3465216	3465216
1	50000	11018918	3624960	3624960
2	50000	12750523	3616768	3616768
3	50000	10472972	3620864	3620864
4	50000	11112507	3620864	3620864
5	50000	10938495	3629056	3629056
Average:		11258683	3622502.4	3622502.4

SLEEP RESULTS

SLEEP SORT (Partially)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY(bytes)	PEAK MEMORY(bytes)
1	1000	3422	4636672	20316160
2	1000	3421	4747264	20312064
3	1000	3457	4960256	20357120
4	1000	3412	4792320	20164608
5	1000	3434	4694016	20287488
Average:		3429.2	4766105.6	20287488
1	10000	35157	12619776	168644608
2	10000	35248	12218368	167530496
3	10000	35251	11984896	167124992
4	10000	35148	12107776	167976960
5	10000	35125	12148736	168087552
Average:		35185.8	12215910.4	167872921.6
1	15000	106449	25128960	493666304
2	15000	106437	27455488	493830144
3	15000	106644	25370624	493236224
4	15000	106479	25542656	494194688
5	15000	106409	25468928	494358528
Average:		106483.6	25793331.2	493857177.6
1	30000	186679	28041216	163287040
2	30000	188399	39071744	173277184
3	30000	184115	38809600	172703744
4	30000	170428	42221568	182853632
5	30000	170880	34516992	173260800
Average:		180100.2	36532224	173076480

SLEEP SORT (Random)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY(bytes)	PEAK MEMORY(bytes)
1	1000	6630	4489216	13103104
2	1000	7055	4575232	13914112
3	1000	6512	4407296	12230856
4	1000	6481	4419584	12521472
5	1000	6536	4460544	12931072
Average:		6642.8	4470374.4	12940083.2
1	10000	94850	15269888	97054720
2	10000	96624	35897344	100630528
3	10000	95999	17031168	99790848
4	10000	98199	17432576	102629376
5	10000	101580	17670144	103944192
Average:		97450.4	20660224	100809932.8
1	15000	160145	21856256	137322496
2	15000	129462	27414528	149745664
3	15000	121057	28971008	146456576
4	15000	118856	27021312	138620928
5	15000	121736	26255360	141905920
Average:		130251.2	26303692.8	142810316.8
1	30000	365969	36884480	214315008
2	30000	365706	38985728	216928256
3	30000	368807	37588992	218697728
4	30000	383965	33177600	222220288
5	30000	364588	29302784	203378688
Average:		369807	35187916.8	215107993.6

COMPARISON

PARTIALLY

- Cocktail wins in terms of run time

RANDOM

- Sleep sort wins in terms of run time and memory.
- Cocktail win in terms of memory

TIME COMPLEXITY

- Stalin best case: $O(N)$
- Stalin worst case: $O(N^2)$
- Cocktail best case: $O(N)$
- Cocktail worst case: $O(N^2)$
- Sleep sort: $O(N\log N + \max(\text{input}))$

CODE SIMPLICITY

1. Safe Stalin
2. Sleep sort
3. Cocktail Shaker

CONCLUSION

- Cocktail Shaker sort has the fastest run time and is the most memory efficient
- Stalin sort surprisingly has the longest run time
- Sleep sort takes the most memory

01

The original stalin sort deletes the unsorted element and thus cannot be applied to sort any arrays correctly.

02

No functional use for sleep sort as it sleeps based on the value of x element in the array.



THANK YOU



o o o o

<https://github.com/Pandalmation/Strange-Sorting-Algorithms>