STRANGE ALGORITHMS COMPARISON

ADA FINAL PROJECT

Final Presentation

Jocelin | Nicholas | Tiffany

TABLE OF CONTENTS

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

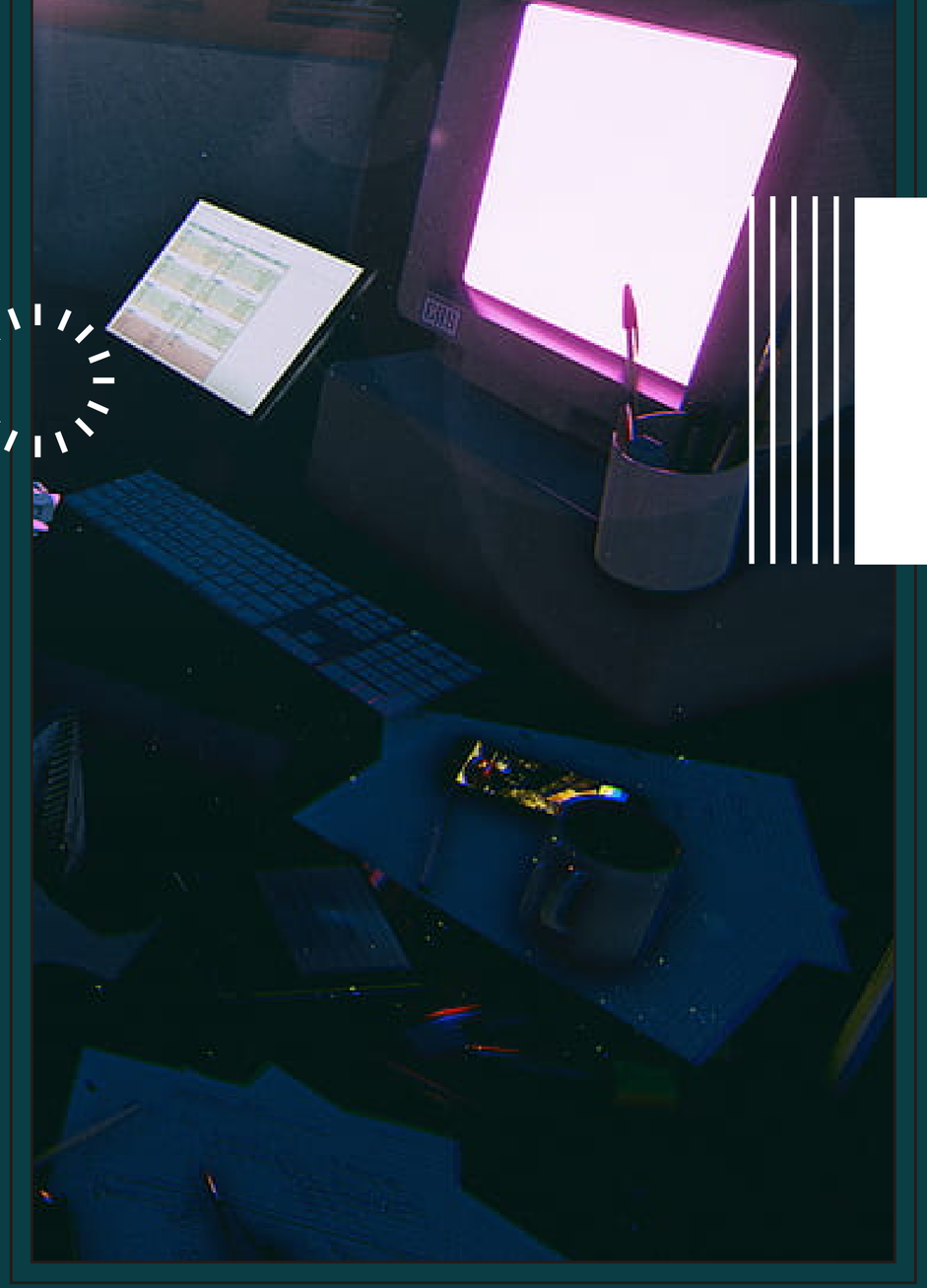
BACKGROUND

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

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

PROCESS 04

Identifying pros and cons of each algorithm

PROCESS 05

Identifying real life samples of when such algorithm is used

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	4186112	4190208
5	1000	329	4186112	4190208
	Average:	355	4193484.8	4197580.8
1	10000	26426	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:	601084.4	4381900.8	4385996.8

COCKTAIL RESULTS

COCKTAIL SORT (Partially)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY	PEAK MEMORY
1	1000	0.028	3956736	3960832
2	1000	0.033	3964928	3969024
3	1000	0.04	3952640	3956736
4	1000	0.028	3956736	3960832
5	1000	0.037	3969024	3973120
	Average:	0.0332	3960012.8	3964108.8
1	10000	0.508	4083712	4087808
2	10000	0.438	4055040	4059136
3	10000	0.448	4075520	4079616
4	10000	0.57	4046848	4050944
5	10000	0.672	4087808	4091904
	Average:	0.5272	4069785.6	4073881.6
1	15000	0.442	4157440	4161536
2	15000	0.875	4141056	4145152
3	15000	0.942	4149248	415344
4	15000	0.672	4153344	4157440
5	15000	1.135	4136960	4141056
	Average:	0.8132	4147609.6	3404105.6
1	30000	1.44	4349952	4354048
2	30000	1.272	4358144	4362240
3	30000	1.861	4349952	4354048
4	30000	1.284	4313088	4317184
5	30000	1.355	4354048	4358144
	Average:	1.4424	4345036.8	4349132.8
1	50000	2.119	4636672	4640768
2	50000	2.604	4632576	4636672
3	50000	2.527	4595712	4599808
4	50000	2.661	4628480	4632576
5	50000	2.741	4616192	4620288
	Average:	2.5304	4621926.4	4626022.4

COCKTAIL SORT (Random)				
RUNS	SIZE	RUNTIME(MS)	CURRENT MEMORY	PEAK MEMORY
1	1000	2.428	3371008	3375104
2	1000	2.419	3379200	3383296
3	1000	2.415	3383296	3387392
4	1000	2.442	3387392	3391488
5	1000	2.491	3383296	3387392
	Average:	2.439	3380838.4	3384934.4
1	10000	8.789	3969024	3973120
2	10000	8.636	3956736	3960832
3	10000	7.664	3952640	3956736
4	10000	8.387	3952640	3956736
5	10000	8.672	3956736	3960832
	Average:	8.4296	3957555.2	3961651.2
1	15000	632	3493888	3497984
2	15000	708	3489792	3493888
3	15000	602	3497984	3497984
4	15000	603	3493888	3497984
5	15000	615	3489792	3493888
	Average:	632	3493068.8	3496345.6
1	30000	2507	3624960	3629056
2	30000	2545	3592192	3596288
3	30000	2695	3612672	3616768
4	30000	2473	3588096	3592192
5	30000	2449	3616768	3620864
	Average:	2533.8	3606937.6	3611033.6
1	50000	7139	3751936	3756032
2	50000	7027	3743744	3747840
3	50000	6985	3764224	3768320
4	50000	7327	3768320	3772416
5	50000	7311	3768320	3772416
	Average:	7157.8	3759308.8	3763404.8

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	1040	4751360	20099072
2	1000	1040	4718592	20062208
3	1000	1046	4730880	20115456
4	1000	1043	4763648	20217856
5	1000	1044	4694016	20279296
	Average:	1042.6	4731699.2	20154777.6
1	10000	3204	4792320	15507456
2	10000	3443	4722688	14962688
3	10000	3154	4730880	14974976
4	10000	3187	4657152	14372864
5	10000	3221	4767744	15237120
	Average:	3241.8	4734156.8	15011020.8
1	15000	16034	15544320	236912640
2	15000	15773	16207872	245575680
3	15000	16087	15568896	238067712
4	15000	15742	15355904	245161984
5	15000	15974	16146432	240152576
	Average:	15922	15764684.8	241174118.4
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
- Stalin sort wins in terms of run time

RANDOM

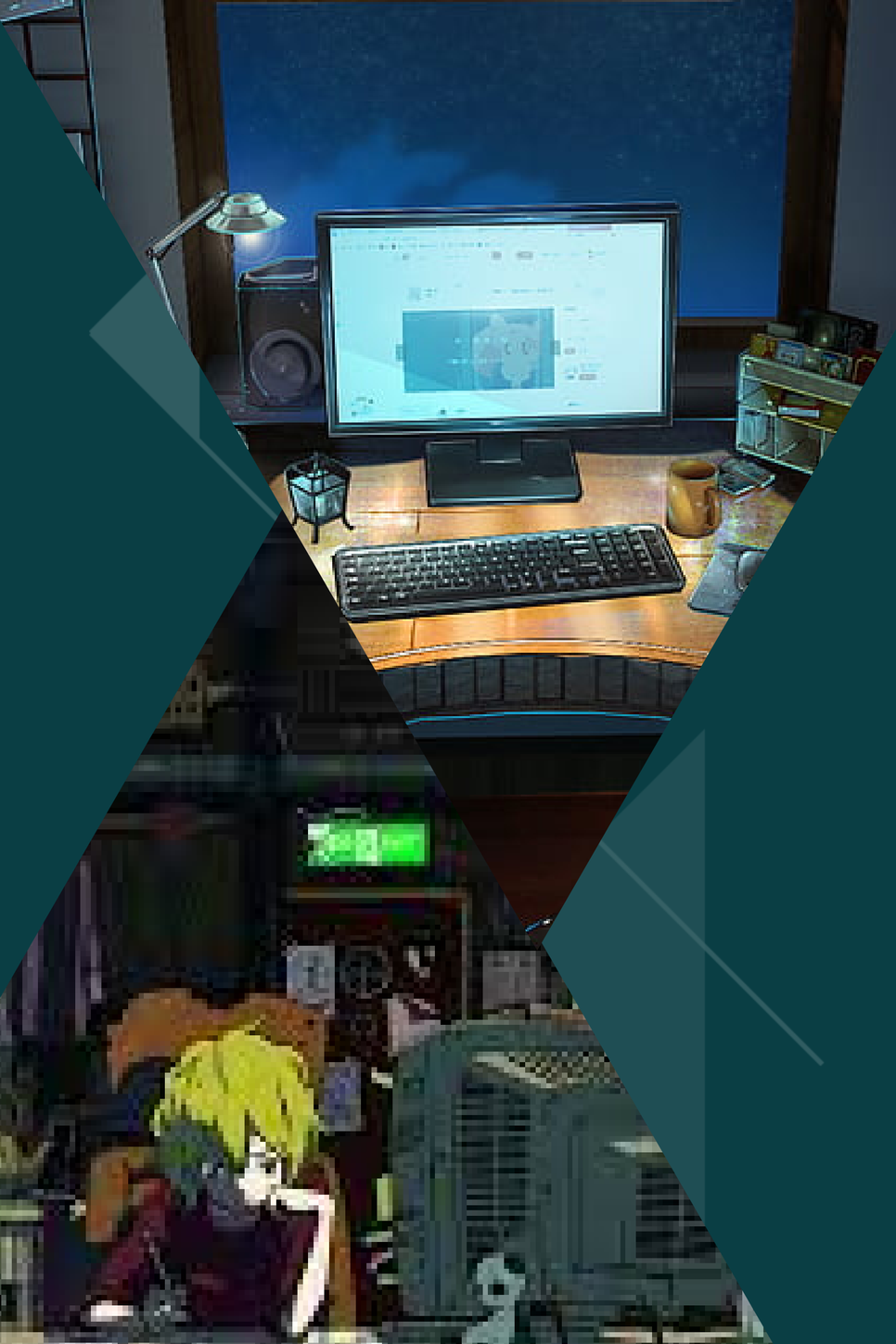
- Sleep sort wins in terms of run time and memory.
- Cocktail win in terms of memory

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(NlogN + max(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