

# **ENAE 380: Lab 02**

Due on September 29, 2024 at 11:59 PM

*Dr. Mumu Xu, 0106*

**Vai Srivastava**

September 28, 2024

## Problem 1

### 6.1.1.1 Bubble Sort:

Implements the Bubble Sort algorithm to sort a list of integers in ascending order. It repeatedly steps through the list, compares adjacent elements, and swaps them if they are in the wrong order. The process continues until no more swaps are needed, indicating that the list is sorted.

## Problem 2

### 6.1.1.2 Selection Sort:

Implements the Selection Sort algorithm to sort a list of integers in ascending order. The function divides the list into a sorted and unsorted part. It repeatedly selects the smallest element from the unsorted portion and swaps it with the first unsorted element, expanding the sorted portion one element at a time.

## Problem 3

### 6.1.1.3 Insertion Sort:

Implements the Insertion Sort algorithm to sort a list of integers in ascending order. It builds the final sorted list one item at a time. For each element in the list, it compares it with the elements before it and inserts it into its correct position within the sorted portion of the list.

## Problem 4

### 6.1.1.4 Merge Sort:

Implements the Merge Sort algorithm to sort a list of integers in ascending order. The function uses a divide-and-conquer approach: it recursively splits the list into halves until single-element lists are obtained, then merges these smaller lists back together in sorted order to form the final sorted list.

## Problem 5

### 6.1.2 Sorting Test:

Runs performance tests on the different sorting algorithms (Bubble Sort, Selection Sort, Insertion Sort, and Merge Sort) using arrays of varying sizes. It measures and records the execution times of each algorithm for arrays of sizes [8, 200, 500, 1000, 10000]. The results are printed to the console and saved to a CSV file named "SortingTests.csv" for further analysis.

The produced table can be viewed at the end of the document.

Through simple observation of the table, we can see that Merge Sort is by far the most time-efficient of these sorting functions. The average execution time of each function across all 500 trials is shown in the table below:

Table 1: Average Execution Time of Sorting Algorithms

Algorithm	Execution Time [s]
Bubble Sort	0.604490
Selection Sort	0.234296
Insertion Sort	0.232649
Merge Sort	0.002582

This makes a great deal of sense when considering the known information about sorting algorithm time and space complexity. The following table displays this information:

Table 2: Complexity of Sorting Algorithms

Algorithm	Average Time Complexity	Worst Case Space Complexity
Bubble Sort	$O(n^2)$	$O(1)$
Selection Sort	$O(n^2)$	$O(1)$
Insertion Sort	$O(n^2)$	$O(1)$
Merge Sort	$O(n \log(n))$	$O(n)$

## Problem 6

### 6.2 Read File:

Reads a file containing numbers (one per line), sorts them using the Merge Sort algorithm, and writes the sorted numbers to a new file called "SortedNumbers.txt". The output file includes a header "Sorted Numbers" followed by the sorted numbers, each on a new line. This function skips the first line of the input file, assuming it is a header.

## Problem 7

### 6.3.1 Vigenere Cipher:

Encodes a plaintext message using the Vigenère cipher with a user-provided keyword. The function prompts the user to input a plaintext message and a keyword. It then shifts each alphabetic character in the message by an amount determined by the corresponding character in the keyword. Non-alphabetic characters are left unchanged. The encoded message is printed in lowercase.



## Problem 8

### 6.3.2 Decrypt:

Attempts to decrypt a Vigenère cipher text read from a file named "cipher.txt" by trying every 5-letter word from "validwords.txt" as a potential keyword. For each keyword, it decrypts the cipher text and checks if the first 5-letter word in the decrypted message is a valid word from the dictionary. If a valid decryption is found, it displays the decrypted message and asks the user to confirm whether it looks correct. If confirmed, it prints the valid decryption and returns the decrypted message.

## Problem 9

### 6.4 Tortoise and the Hare:

#### Summary of Functions and Classes

##### 1. `race(self)`

**Purpose:** Simulates a real-time race between two runners—a tortoise and a hare—and displays the race progress on the console. The race continues until one of the runners reaches the finish line, and the winner is announced at the end.

**Detailed Description:**

- **Initialization:**

- Creates two **Runner** objects representing the tortoise and the hare, each with different speeds and visual representations:
  - \* Tortoise: speed of 2 units per turn, visualized by a turtle emoji.
  - \* Hare: speed of 8 units per turn, visualized by a rabbit emoji.
- Initializes a list **finished** to track whether any racer has completed the race.
- Defines the visual representation of the racetrack using Unicode box-drawing characters:
  - \* **track\_top**: the top border of the track.
  - \* **track\_bot**: the bottom border of the track.
- Clears the console using the `clear()` function to prepare for the race countdown.

- **Race Countdown:**

- Performs a countdown from 3 to 1 with a 1-second interval between each number, displaying "3...", "2...", "1...".
- After the countdown, prints "GO!" to signal the start of the race.

- **Race Loop:**

- Enters a loop that continues until one of the runners reaches the finish line (when their visual length exceeds 50 characters).
- Inside the loop:
  - \* Pauses for 1 second to simulate real-time progression.
  - \* Clears the console to update the race display.
  - \* Prints the racetrack's top border.
  - \* Updates each racer's position:
    - Checks if the racer's visual length is less than 50 (i.e., they haven't finished).
    - Generates a random number between 1 and the racer's speed to simulate movement.
    - Prepends a number of spaces to the racer's visual representation to move them forward on the track.
    - Prints the racer's updated position on the track.
  - \* Prints the racetrack's bottom border.
  - \* Updates the **finished** list to check if any racer has completed the race.

- **Determining the Winner:**

- After the loop ends, determines which racer has won based on who finished first.
- Prints a congratulatory message announcing the winner (either "tortoise" or "hare").

## 2. `clear()`

**Purpose:** Clears the console screen to refresh the display during the race simulation.

**Detailed Description:**

- Uses ANSI escape codes to reset the terminal and clear the scrollbar buffer:
  - `\033c`: Resets the terminal to its initial state.
  - `\033[3J`: Clears the entire screen and the scrollbar buffer.
- The `end=""` parameter in the `print()` function ensures that no additional newline is added after clearing the console.

## 3. `class Animal`

**Purpose:** Represents a basic animal with attributes for name, color, and a visual representation. Serves as a base class for other animal types.

**Attributes:**

- `name (str)`: The name of the animal.
- `color (str)`: The color of the animal.
- `visual (str)`: A visual representation of the animal using ASCII characters. Defaults to `----{ , _ , '>' }`, which depicts a mouse.

**Methods:**

- `__init__(self, name, color, visual="----{ , _ , '>' }")`:
  - Initializes an `Animal` instance with the provided name, color, and optional visual.
  - Sets the instance attributes `self.name`, `self.color`, and `self.visual`.
- `speak(self)`:
  - Returns a generic greeting string `"hello"`.
  - This method can be overridden by subclasses to provide specific animal sounds.

#### 4. class Runner(Animal)

**Purpose:** Represents a runner in the race, inheriting from the **Animal** class and adding a **speed** attribute. This class models animals that can participate in the race simulation.

**Attributes:**

- Inherits all attributes from **Animal** (**name**, **color**, **visual**).
- **speed** (**int**): The speed of the runner, determining how many spaces they move forward each turn.

**Methods:**

- **\_\_init\_\_(self, name, color, speed, visual):**
  - Calls the parent **Animal** class constructor to initialize **name**, **color**, and **visual**.
  - Initializes the **speed** attribute specific to the **Runner** class.
  - Sets the instance attribute **self.speed**.

**Note:** The **race** function relies on the **Runner** and **Animal** classes to create the racers. The **clear()** function is used within the **race** function to update the console display dynamically, giving the effect of an animated race. The random movement within the racer's speed range adds unpredictability to the race outcome, making it possible for either the tortoise or the hare to win, despite their assigned speeds.

### Overall Summary

- The code simulates a simple console-based race between two animals—a tortoise and a hare—using object-oriented programming concepts in Python.
- The **Animal** class provides a basic structure for animals with common attributes and methods.
- The **Runner** class extends **Animal** to include the **speed** attribute, which is essential for the race simulation.
- The **race** function orchestrates the race, updating and displaying the positions of the runners in real-time until one of them wins.
- The **clear** function enhances the user experience by refreshing the console display, making the race appear animated.

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
0	0.000004	0.000003	1.668930e-06	0.000009
1	0.000810	0.000430	3.240108e-04	0.000158
2	0.006326	0.003258	2.596140e-03	0.000440
3	0.026853	0.011535	1.066685e-02	0.000888
4	2.952146	1.144153	1.126282e+00	0.011721
5	0.000110	0.000002	1.907349e-06	0.000004
6	0.000733	0.000305	3.118515e-04	0.000138
7	0.005761	0.002995	2.301931e-03	0.000414
8	0.026644	0.011564	1.080990e-02	0.000911
9	2.957679	1.151939	1.130784e+00	0.011380
10	0.000107	0.000002	9.536743e-07	0.000005
11	0.000730	0.000305	2.989769e-04	0.000136
12	0.005538	0.002944	2.261162e-03	0.000410
13	0.026567	0.011871	1.102591e-02	0.000917
14	2.973529	1.146492	1.144467e+00	0.011352
15	0.000107	0.000002	1.192093e-06	0.000005
16	0.000723	0.000306	3.006458e-04	0.000137
17	0.005506	0.002971	2.135038e-03	0.000408
18	0.026265	0.011503	1.068330e-02	0.000886
19	2.961063	1.141604	1.142982e+00	0.011374
20	0.000116	0.000007	1.907349e-06	0.000005
21	0.000722	0.000306	2.901554e-04	0.000131
22	0.005654	0.003043	2.346039e-03	0.000415
23	0.025876	0.011499	1.019096e-02	0.000891
24	2.968019	1.140812	1.155751e+00	0.011421
25	0.000110	0.000003	1.907349e-06	0.000004
26	0.000733	0.000303	3.180504e-04	0.000137
27	0.005506	0.002988	2.120256e-03	0.000415
28	0.026019	0.011490	1.063013e-02	0.000879
29	2.972339	1.142940	1.144731e+00	0.011382
30	0.000105	0.000002	1.907349e-06	0.000004
31	0.000734	0.000303	3.008842e-04	0.000138
32	0.005619	0.002970	2.246857e-03	0.000406
33	0.025783	0.011554	9.971857e-03	0.000882
34	2.993282	1.142049	1.143382e+00	0.011365
35	0.000109	0.000002	2.145767e-06	0.000004
36	0.000718	0.000305	2.908707e-04	0.000136
37	0.005608	0.002984	2.247095e-03	0.000411
38	0.025889	0.011465	1.033497e-02	0.000884
39	2.991613	1.152649	1.161297e+00	0.011405
40	0.000116	0.000003	1.907349e-06	0.000005
41	0.000719	0.000303	3.082752e-04	0.000132
42	0.005388	0.002927	2.115011e-03	0.000410
43	0.026439	0.011513	1.080108e-02	0.000879
44	2.970475	1.144538	1.131909e+00	0.011338
45	0.000117	0.000003	1.907349e-06	0.000005
46	0.000751	0.000305	3.378391e-04	0.000134
47	0.005735	0.002991	2.346039e-03	0.000419
48	0.026507	0.011445	1.086426e-02	0.000872
49	2.973246	1.144207	1.145549e+00	0.011343
50	0.000111	0.000003	1.907349e-06	0.000005
51	0.000725	0.000305	2.980232e-04	0.000134

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
52	0.005556	0.002961	2.231121e-03	0.000410
53	0.025944	0.011552	1.034689e-02	0.000871
54	2.957843	1.142411	1.129121e+00	0.011376
55	0.000120	0.000003	1.192093e-06	0.000005
56	0.000707	0.000307	2.830029e-04	0.000134
57	0.005687	0.002938	2.345085e-03	0.000415
58	0.026031	0.011571	1.032209e-02	0.000893
59	2.959556	1.154956	1.151020e+00	0.011381
60	0.000111	0.000002	2.145767e-06	0.000004
61	0.000731	0.000306	3.011227e-04	0.000134
62	0.005606	0.002976	2.256870e-03	0.000415
63	0.026148	0.011502	1.048779e-02	0.000888
64	2.948929	1.141785	1.133863e+00	0.011967
65	0.000106	0.000002	2.145767e-06	0.000005
66	0.000717	0.000303	3.142357e-04	0.000133
67	0.005680	0.002988	2.255678e-03	0.000434
68	0.025866	0.011488	1.037288e-02	0.000884
69	2.972906	1.153612	1.148173e+00	0.011357
70	0.000114	0.000003	2.145767e-06	0.000005
71	0.000702	0.000304	2.927780e-04	0.000135
72	0.005596	0.003005	2.233982e-03	0.000413
73	0.026151	0.011474	1.052999e-02	0.000883
74	2.960084	1.155793	1.139690e+00	0.011407
75	0.000118	0.000003	1.192093e-06	0.000005
76	0.000712	0.000304	2.880096e-04	0.000134
77	0.005651	0.003036	2.282143e-03	0.000410
78	0.025876	0.011520	1.026225e-02	0.000895
79	2.949237	1.153273	1.144767e+00	0.011382
80	0.000108	0.000003	7.152557e-07	0.000005
81	0.000718	0.000304	2.989769e-04	0.000134
82	0.005670	0.002933	2.403021e-03	0.000418
83	0.026111	0.011525	1.043200e-02	0.000886
84	2.958412	1.141500	1.137442e+00	0.011346
85	0.000119	0.000003	1.907349e-06	0.000005
86	0.000728	0.000303	3.082752e-04	0.000135
87	0.005610	0.003007	2.243042e-03	0.000411
88	0.025957	0.011505	1.049280e-02	0.000892
89	2.982551	1.154004	1.155007e+00	0.011378
90	0.000106	0.000002	2.145767e-06	0.000004
91	0.000716	0.000305	3.068447e-04	0.000135
92	0.005648	0.002961	2.274990e-03	0.000408
93	0.026082	0.011509	1.044393e-02	0.000881
94	2.964234	1.142216	1.140515e+00	0.011337
95	0.000125	0.000003	1.907349e-06	0.000005
96	0.000728	0.000304	2.958775e-04	0.000134
97	0.005654	0.003265	2.532244e-03	0.000411
98	0.026171	0.011491	1.133585e-02	0.000949
99	2.964949	1.143243	1.132573e+00	0.011349
100	0.000123	0.000003	9.536743e-07	0.000005
101	0.000719	0.000305	3.030300e-04	0.000137
102	0.005689	0.002999	2.316952e-03	0.000416
103	0.026521	0.011528	1.065135e-02	0.000889

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
104	2.966545	1.151094	1.147261e+00	0.011369
105	0.000124	0.000004	9.536743e-07	0.000005
106	0.000752	0.000310	3.271103e-04	0.000132
107	0.005610	0.003000	2.295732e-03	0.000433
108	0.026064	0.011477	1.028299e-02	0.000884
109	2.938943	1.151706	1.127634e+00	0.011417
110	0.000132	0.000003	1.907349e-06	0.000005
111	0.000719	0.000302	2.939701e-04	0.000135
112	0.005463	0.002942	2.158880e-03	0.000412
113	0.026380	0.011548	1.051593e-02	0.000876
114	2.974675	1.144197	1.131282e+00	0.011350
115	0.000120	0.000003	9.536743e-07	0.000004
116	0.000670	0.000303	2.601147e-04	0.000139
117	0.005612	0.002999	2.238989e-03	0.000411
118	0.026187	0.011514	1.048994e-02	0.000887
119	2.974847	1.152631	1.154915e+00	0.011381
120	0.000120	0.000003	2.145767e-06	0.000005
121	0.000726	0.000302	3.073215e-04	0.000134
122	0.005543	0.002962	2.195835e-03	0.000413
123	0.026399	0.011449	1.079822e-02	0.000881
124	2.972861	1.140614	1.129256e+00	0.011335
125	0.000117	0.000003	2.145767e-06	0.000004
126	0.000702	0.000305	2.770424e-04	0.000136
127	0.005661	0.002926	2.251148e-03	0.000409
128	0.026811	0.011418	1.112986e-02	0.000882
129	2.973025	1.142824	1.141346e+00	0.011366
130	0.000119	0.000002	1.192093e-06	0.000005
131	0.000752	0.000302	3.211498e-04	0.000132
132	0.005669	0.002956	2.269030e-03	0.000412
133	0.025720	0.011442	1.010418e-02	0.000880
134	2.958281	1.152861	1.148030e+00	0.011411
135	0.000113	0.000002	2.145767e-06	0.000004
136	0.000709	0.000304	2.970695e-04	0.000135
137	0.005603	0.002958	2.243996e-03	0.000413
138	0.026522	0.011523	1.089191e-02	0.000888
139	2.957121	1.145323	1.136053e+00	0.011365
140	0.000137	0.000002	1.907349e-06	0.000004
141	0.000719	0.000303	3.011227e-04	0.000133
142	0.005665	0.002968	2.291918e-03	0.000415
143	0.025973	0.011475	1.037002e-02	0.000881
144	2.945393	1.167009	1.141908e+00	0.011427
145	0.000120	0.000003	1.907349e-06	0.000005
146	0.000712	0.000301	2.892017e-04	0.000135
147	0.005644	0.002945	2.295971e-03	0.000412
148	0.026355	0.011497	1.069736e-02	0.000889
149	2.945746	1.152447	1.145014e+00	0.011408
150	0.000123	0.000003	1.907349e-06	0.000005
151	0.000743	0.000308	3.199577e-04	0.000131
152	0.005625	0.002937	2.281904e-03	0.000409
153	0.026703	0.011864	1.096511e-02	0.000913
154	2.951096	1.148315	1.139698e+00	0.011324
155	0.000118	0.000003	2.145767e-06	0.000004
156	0.000734	0.000302	3.230572e-04	0.000132

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
157	0.005634	0.002957	2.247095e-03	0.000412
158	0.026042	0.011484	1.045299e-02	0.000884
159	2.963182	1.153769	1.150575e+00	0.011376
160	0.000111	0.000003	1.907349e-06	0.000005
161	0.000714	0.000302	2.911091e-04	0.000132
162	0.005651	0.002997	2.258062e-03	0.000412
163	0.026327	0.011506	1.061130e-02	0.000879
164	2.951103	1.142133	1.130321e+00	0.011380
165	0.000124	0.000003	2.145767e-06	0.000005
166	0.000700	0.000305	2.961159e-04	0.000134
167	0.005746	0.002984	2.341986e-03	0.000409
168	0.025767	0.011718	1.057196e-02	0.000909
169	2.959025	1.146619	1.139795e+00	0.011422
170	0.000115	0.000003	2.145767e-06	0.000005
171	0.000737	0.000302	2.911091e-04	0.000131
172	0.005460	0.002906	2.140999e-03	0.000411
173	0.026845	0.011501	1.108408e-02	0.000884
174	2.961400	1.140404	1.130586e+00	0.011363
175	0.000114	0.000003	9.536743e-07	0.000005
176	0.000726	0.000305	3.039837e-04	0.000135
177	0.005686	0.002962	2.310991e-03	0.000411
178	0.026428	0.011544	1.080179e-02	0.000885
179	2.955005	1.154672	1.147207e+00	0.011411
180	0.000115	0.000003	2.145767e-06	0.000005
181	0.000726	0.000302	2.970695e-04	0.000137
182	0.005761	0.003016	2.371788e-03	0.000410
183	0.026043	0.011575	1.036382e-02	0.000886
184	2.995445	1.141891	1.156240e+00	0.011346
185	0.000110	0.000002	1.668930e-06	0.000004
186	0.000742	0.000302	3.180504e-04	0.000134
187	0.005493	0.003001	2.185106e-03	0.000411
188	0.026269	0.011560	1.049900e-02	0.000892
189	2.956258	1.151095	1.156363e+00	0.011388
190	0.000122	0.000003	1.907349e-06	0.000005
191	0.000742	0.000306	3.170967e-04	0.000133
192	0.005678	0.002964	2.243996e-03	0.000420
193	0.026001	0.011519	1.048088e-02	0.000883
194	2.963905	1.146073	1.135526e+00	0.011344
195	0.000120	0.000002	1.907349e-06	0.000004
196	0.000729	0.000303	3.042221e-04	0.000135
197	0.005622	0.002914	2.319098e-03	0.000410
198	0.025862	0.011491	1.034498e-02	0.000886
199	2.948345	1.140056	1.156536e+00	0.011416
200	0.000119	0.000003	1.907349e-06	0.000005
201	0.000729	0.000304	3.070831e-04	0.000136
202	0.005758	0.002962	2.428293e-03	0.000415
203	0.025910	0.011696	1.075292e-02	0.000890
204	2.947653	1.152372	1.142261e+00	0.011398
205	0.000120	0.000003	1.907349e-06	0.000005
206	0.000736	0.000304	3.168583e-04	0.000134
207	0.005759	0.002933	2.351999e-03	0.000410
208	0.026441	0.011514	1.068497e-02	0.000887



	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
209	2.956386	1.154756	1.141059e+00	0.011607
210	0.000111	0.000002	1.907349e-06	0.000004
211	0.000724	0.000304	2.992153e-04	0.000131
212	0.005734	0.002929	2.327204e-03	0.000411
213	0.030055	0.011486	1.074791e-02	0.000891
214	2.973909	1.148708	1.236469e+00	0.011346
215	0.000121	0.000002	1.907349e-06	0.000004
216	0.000727	0.000302	3.011227e-04	0.000135
217	0.005677	0.002972	2.324104e-03	0.000408
218	0.026187	0.011559	1.077199e-02	0.000898
219	2.962655	1.164057	1.147044e+00	0.011396
220	0.000128	0.000004	9.536743e-07	0.000004
221	0.000722	0.000306	2.980232e-04	0.000134
222	0.005675	0.002972	2.316952e-03	0.000431
223	0.025953	0.011464	1.045299e-02	0.000882
224	2.946756	1.151857	1.123563e+00	0.011353
225	0.000125	0.000004	9.536743e-07	0.000004
226	0.000716	0.000307	2.989769e-04	0.000134
227	0.005509	0.002951	2.167225e-03	0.000413
228	0.025526	0.011458	1.014185e-02	0.000885
229	2.959827	1.153864	1.155623e+00	0.011417
230	0.000123	0.000004	7.152557e-07	0.000005
231	0.000751	0.000304	3.030300e-04	0.000134
232	0.005578	0.002942	2.238989e-03	0.000409
233	0.026280	0.011447	1.070094e-02	0.000882
234	2.965898	1.144136	1.142978e+00	0.011365
235	0.000122	0.000003	9.536743e-07	0.000005
236	0.000724	0.000307	3.008842e-04	0.000135
237	0.005542	0.002979	2.170086e-03	0.000413
238	0.026112	0.011496	1.043797e-02	0.000884
239	2.970546	1.140379	1.141913e+00	0.011321
240	0.000126	0.000003	2.145767e-06	0.000004
241	0.000714	0.000302	2.739429e-04	0.000134
242	0.005610	0.002986	2.252102e-03	0.000411
243	0.026127	0.011533	1.033115e-02	0.000894
244	2.964182	1.142192	1.137790e+00	0.011391
245	0.000120	0.000002	9.536743e-07	0.000004
246	0.000721	0.000305	3.049374e-04	0.000137
247	0.005619	0.002959	2.282858e-03	0.000411
248	0.026232	0.011870	1.086211e-02	0.000914
249	2.935134	1.143262	1.119489e+00	0.011676
250	0.000110	0.000002	1.907349e-06	0.000005
251	0.000724	0.000311	3.099442e-04	0.000139
252	0.005758	0.002906	2.350092e-03	0.000413
253	0.027699	0.012409	1.556897e-02	0.000889
254	2.980527	1.141222	1.129145e+00	0.011319
255	0.000113	0.000004	3.099442e-06	0.000005
256	0.000710	0.000302	2.779961e-04	0.000134
257	0.005736	0.002996	2.279997e-03	0.000410
258	0.026088	0.011467	1.045704e-02	0.000886
259	2.997055	1.140736	1.148826e+00	0.011350
260	0.000112	0.000002	9.536743e-07	0.000004
261	0.000755	0.000306	3.478527e-04	0.000136

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
262	0.005614	0.002985	2.230167e-03	0.000413
263	0.026588	0.011484	1.086593e-02	0.000889
264	2.954503	1.145008	1.156173e+00	0.011339
265	0.000111	0.000003	2.145767e-06	0.000005
266	0.000720	0.000305	2.906322e-04	0.000131
267	0.005656	0.003230	2.373934e-03	0.000439
268	0.026282	0.011481	1.051784e-02	0.000884
269	2.966777	1.142176	1.136815e+00	0.011380
270	0.000113	0.000003	1.907349e-06	0.000005
271	0.000720	0.000304	2.901554e-04	0.000133
272	0.005624	0.002948	2.300024e-03	0.000432
273	0.025928	0.011551	1.029992e-02	0.000876
274	2.998735	1.146934	1.137028e+00	0.011442
275	0.000116	0.000003	9.536743e-07	0.000004
276	0.000710	0.000301	2.889633e-04	0.000138
277	0.005677	0.003008	2.295017e-03	0.000410
278	0.025742	0.011467	1.023984e-02	0.000886
279	2.979595	1.173887	1.168402e+00	0.011665
280	0.000108	0.000002	1.907349e-06	0.000004
281	0.000750	0.000303	3.020763e-04	0.000133
282	0.005783	0.002931	2.423048e-03	0.000412
283	0.027126	0.011602	1.093602e-02	0.000889
284	3.002447	1.169948	1.161456e+00	0.011404
285	0.000126	0.000004	9.536743e-07	0.000005
286	0.000728	0.000301	3.042221e-04	0.000131
287	0.005852	0.002929	2.496004e-03	0.000410
288	0.026471	0.011523	1.081395e-02	0.000885
289	2.984028	1.155518	1.133299e+00	0.011334
290	0.000115	0.000002	9.536743e-07	0.000004
291	0.000744	0.000302	3.159046e-04	0.000131
292	0.005571	0.002986	2.250195e-03	0.000410
293	0.025840	0.011676	1.072121e-02	0.000947
294	2.973507	1.165743	1.148948e+00	0.011405
295	0.000113	0.000002	1.907349e-06	0.000004
296	0.000721	0.000308	3.030300e-04	0.000136
297	0.005701	0.002942	2.273083e-03	0.000409
298	0.026201	0.011494	1.063514e-02	0.000883
299	3.009878	1.148242	1.151362e+00	0.011372
300	0.000117	0.000002	9.536743e-07	0.000005
301	0.000722	0.000306	2.961159e-04	0.000135
302	0.005681	0.002988	2.309799e-03	0.000410
303	0.026465	0.011552	1.076293e-02	0.000884
304	3.017558	1.157472	1.173935e+00	0.011586
305	0.000114	0.000002	1.907349e-06	0.000004
306	0.000739	0.000303	3.230572e-04	0.000137
307	0.005744	0.003001	2.328873e-03	0.000431
308	0.026166	0.011729	1.077199e-02	0.000891
309	3.002417	1.158786	1.160349e+00	0.011420
310	0.000116	0.000002	1.192093e-06	0.000005
311	0.000813	0.000306	2.999306e-04	0.000137
312	0.005643	0.002918	2.242088e-03	0.000420
313	0.026767	0.011599	1.095676e-02	0.000885

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
314	3.031846	1.160577	1.154089e+00	0.011376
315	0.000114	0.000002	2.145767e-06	0.000004
316	0.000718	0.000306	2.939701e-04	0.000134
317	0.005765	0.002969	2.299070e-03	0.000413
318	0.026360	0.011469	1.065898e-02	0.000882
319	2.968847	1.153868	1.132835e+00	0.011349
320	0.000120	0.000003	1.907349e-06	0.000005
321	0.000736	0.000302	3.139973e-04	0.000134
322	0.005752	0.002951	2.315998e-03	0.000412
323	0.025976	0.011535	1.070499e-02	0.000896
324	3.029331	1.148188	1.160684e+00	0.011565
325	0.000120	0.000002	9.536743e-07	0.000004
326	0.000726	0.000310	2.830029e-04	0.000136
327	0.005689	0.002891	2.286911e-03	0.000412
328	0.026459	0.011426	1.090598e-02	0.000884
329	2.980177	1.152341	1.135211e+00	0.011452
330	0.000114	0.000002	2.145767e-06	0.000004
331	0.000750	0.000303	3.228188e-04	0.000137
332	0.005806	0.003078	2.388954e-03	0.000411
333	0.025752	0.011604	1.014018e-02	0.000875
334	3.011954	1.173149	1.148448e+00	0.011474
335	0.000116	0.000003	1.192093e-06	0.000005
336	0.000742	0.000304	3.151894e-04	0.000134
337	0.005797	0.002951	2.421141e-03	0.000411
338	0.026018	0.011670	1.047492e-02	0.000887
339	2.952113	1.148872	1.133240e+00	0.011337
340	0.000123	0.000003	2.145767e-06	0.000005
341	0.000725	0.000303	3.030300e-04	0.000134
342	0.005679	0.002954	2.274990e-03	0.000413
343	0.026486	0.011469	1.088810e-02	0.000891
344	2.988540	1.148980	1.139890e+00	0.011715
345	0.000104	0.000002	9.536743e-07	0.000005
346	0.000723	0.000312	2.853870e-04	0.000136
347	0.005945	0.002972	2.310991e-03	0.000420
348	0.026402	0.011511	1.063085e-02	0.000878
349	3.023343	1.170654	1.152916e+00	0.011576
350	0.000111	0.000006	1.907349e-06	0.000005
351	0.000722	0.000309	2.770424e-04	0.000132
352	0.005667	0.002983	2.291918e-03	0.000409
353	0.026583	0.011642	1.086593e-02	0.000899
354	2.994936	1.164047	1.163779e+00	0.011574
355	0.000106	0.000002	2.145767e-06	0.000004
356	0.000757	0.000309	2.932549e-04	0.000140
357	0.005742	0.003091	2.383947e-03	0.000431
358	0.026899	0.011792	1.106119e-02	0.000908
359	3.000273	1.154660	1.154754e+00	0.011438
360	0.000134	0.000002	9.536743e-07	0.000004
361	0.000715	0.000304	2.901554e-04	0.000135
362	0.005635	0.003047	2.328157e-03	0.000415
363	0.026362	0.011621	1.083684e-02	0.000915
364	2.989944	1.163094	1.147412e+00	0.011722
365	0.000100	0.000002	1.907349e-06	0.000004
366	0.000731	0.000303	3.039837e-04	0.000132

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
367	0.005696	0.002888	2.251148e-03	0.000406
368	0.026084	0.011583	1.052785e-02	0.000890
369	2.994663	1.150148	1.150992e+00	0.011456
370	0.000108	0.000006	1.192093e-06	0.000005
371	0.000706	0.000303	2.970695e-04	0.000133
372	0.005890	0.002910	2.429962e-03	0.000411
373	0.025739	0.011498	1.021075e-02	0.000909
374	2.973955	1.168414	1.151282e+00	0.011385
375	0.000107	0.000005	2.145767e-06	0.000004
376	0.000744	0.000304	3.108978e-04	0.000135
377	0.005727	0.002975	2.328157e-03	0.000415
378	0.026744	0.011488	1.106691e-02	0.000878
379	2.985720	1.161800	1.155169e+00	0.011442
380	0.000114	0.000002	1.907349e-06	0.000004
381	0.000748	0.000314	3.159046e-04	0.000136
382	0.005642	0.003005	2.252102e-03	0.000413
383	0.026676	0.011574	1.104307e-02	0.000886
384	3.001692	1.172056	1.161593e+00	0.011387
385	0.000119	0.000004	1.668930e-06	0.000005
386	0.000718	0.000320	3.232956e-04	0.000131
387	0.005585	0.002977	2.234221e-03	0.000424
388	0.026034	0.011523	1.035523e-02	0.000889
389	3.000289	1.161550	1.145132e+00	0.011539
390	0.000114	0.000002	2.145767e-06	0.000004
391	0.000743	0.000309	3.371239e-04	0.000139
392	0.005781	0.003010	2.330065e-03	0.000430
393	0.026199	0.011535	1.045299e-02	0.000888
394	2.982838	1.181797	1.172622e+00	0.011536
395	0.000116	0.000003	2.145767e-06	0.000005
396	0.000705	0.000304	3.051758e-04	0.000132
397	0.005681	0.002959	2.298117e-03	0.000411
398	0.026870	0.011648	1.116419e-02	0.000887
399	3.012925	1.163815	1.166928e+00	0.011648
400	0.000139	0.000003	9.536743e-07	0.000005
401	0.000768	0.000304	3.330708e-04	0.000133
402	0.005566	0.002968	2.178192e-03	0.000436
403	0.026963	0.011922	1.086378e-02	0.000993
404	3.006097	1.169701	1.147195e+00	0.011790
405	0.000113	0.000002	2.145767e-06	0.000005
406	0.000793	0.000305	3.268719e-04	0.000133
407	0.005650	0.003023	2.204895e-03	0.000412
408	0.026447	0.011783	1.072001e-02	0.000890
409	3.036960	1.170897	1.160935e+00	0.011575
410	0.000123	0.000008	1.907349e-06	0.000005
411	0.000732	0.000304	3.211498e-04	0.000132
412	0.005654	0.002956	2.310753e-03	0.000411
413	0.026300	0.011560	1.060987e-02	0.000889
414	3.025053	1.169290	1.171695e+00	0.011798
415	0.000119	0.000002	1.668930e-06	0.000004
416	0.000697	0.000303	2.889633e-04	0.000132
417	0.005657	0.003053	2.397776e-03	0.000408
418	0.026629	0.011566	1.096296e-02	0.000889

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
419	3.037614	1.179409	1.172008e+00	0.011470
420	0.000122	0.000004	1.192093e-06	0.000005
421	0.000697	0.000301	2.789497e-04	0.000132
422	0.005696	0.003032	2.284050e-03	0.000413
423	0.025952	0.011649	1.030588e-02	0.000883
424	2.999322	1.191638	1.163154e+00	0.011742
425	0.000124	0.000004	2.145767e-06	0.000006
426	0.000749	0.000309	3.030300e-04	0.000131
427	0.005758	0.002981	2.419949e-03	0.000408
428	0.026236	0.011480	1.092577e-02	0.000887
429	3.014122	1.179499	1.164503e+00	0.011436
430	0.000113	0.000002	9.536743e-07	0.000005
431	0.000738	0.000304	3.190041e-04	0.000134
432	0.005611	0.002917	2.211094e-03	0.000408
433	0.026280	0.011554	1.055408e-02	0.000882
434	3.036792	1.162349	1.186467e+00	0.011614
435	0.000108	0.000002	9.536743e-07	0.000004
436	0.000752	0.000311	3.180504e-04	0.000134
437	0.005633	0.002884	2.193213e-03	0.000411
438	0.026426	0.011596	1.032686e-02	0.000914
439	3.043986	1.165855	1.162665e+00	0.011561
440	0.000120	0.000004	2.145767e-06	0.000005
441	0.000762	0.000309	3.252029e-04	0.000134
442	0.005859	0.002892	2.491236e-03	0.000415
443	0.026225	0.011566	1.065516e-02	0.000883
444	3.047615	1.163389	1.161937e+00	0.011468
445	0.000106	0.000002	1.907349e-06	0.000004
446	0.000715	0.000302	2.839565e-04	0.000131
447	0.005753	0.002921	2.304077e-03	0.000409
448	0.026031	0.011504	1.060987e-02	0.000892
449	3.085794	1.179177	1.170253e+00	0.011618
450	0.000104	0.000003	9.536743e-07	0.000005
451	0.000724	0.000309	2.939701e-04	0.000135
452	0.005778	0.003003	2.357006e-03	0.000411
453	0.026239	0.011497	1.057792e-02	0.000883
454	3.213805	1.200532	1.163525e+00	0.011623
455	0.000107	0.000002	1.907349e-06	0.000004
456	0.000710	0.000309	3.058910e-04	0.000136
457	0.005760	0.002964	2.322912e-03	0.000411
458	0.026626	0.011796	1.055479e-02	0.000887
459	3.021020	1.165663	1.159489e+00	0.011391
460	0.000106	0.000002	9.536743e-07	0.000005
461	0.000721	0.000303	2.942085e-04	0.000133
462	0.005761	0.002896	2.398014e-03	0.000411
463	0.025856	0.011745	1.035690e-02	0.000984
464	3.024332	1.163310	1.155691e+00	0.011492
465	0.000110	0.000002	1.907349e-06	0.000004
466	0.000730	0.000303	3.132820e-04	0.000136
467	0.005654	0.003061	2.221823e-03	0.000418
468	0.026419	0.011607	1.106191e-02	0.000955
469	3.062889	1.197856	1.161736e+00	0.011382
470	0.000107	0.000002	6.914139e-06	0.000006
471	0.000790	0.000303	3.120899e-04	0.000133

	Bubble Sort	Selection Sort	Insertion Sort	Merge Sort
472	0.005560	0.003016	2.189875e-03	0.000413
473	0.025909	0.012020	1.024318e-02	0.000948
474	3.012463	1.162105	1.153458e+00	0.011371
475	0.000114	0.000003	1.907349e-06	0.000004
476	0.000760	0.000305	3.268719e-04	0.000132
477	0.005719	0.002959	2.458811e-03	0.000444
478	0.026136	0.011716	1.026201e-02	0.000881
479	3.067618	1.163030	1.154836e+00	0.011412
480	0.000113	0.000003	1.907349e-06	0.000004
481	0.000762	0.000305	3.457069e-04	0.000163
482	0.005733	0.002952	2.382278e-03	0.000477
483	0.026299	0.011572	1.029611e-02	0.000876
484	3.043707	1.224721	1.159989e+00	0.011783
485	0.000103	0.000002	2.145767e-06	0.000004
486	0.000741	0.000308	2.951622e-04	0.000132
487	0.005626	0.003027	2.388000e-03	0.000452
488	0.026969	0.011858	1.063395e-02	0.000893
489	3.077768	1.175246	1.166123e+00	0.011648
490	0.000118	0.000004	2.145767e-06	0.000005
491	0.000701	0.000310	2.818108e-04	0.000134
492	0.005627	0.002880	2.255201e-03	0.000409
493	0.026351	0.011501	1.076102e-02	0.000899
494	3.079102	1.175422	1.185873e+00	0.011777
495	0.000103	0.000002	1.668930e-06	0.000004
496	0.000750	0.000315	3.201962e-04	0.000138
497	0.006026	0.002988	2.361059e-03	0.000419
498	0.029390	0.011648	1.195288e-02	0.001508
499	3.105427	1.163444	1.168971e+00	0.011363