Course - CVL867
Assignment 1 Solutions
Instructor: Anoop Krishnan

Utkarsh Pratiush
2022SRZ8573

January 20, 2023

Units: Element density: gm/cm^3, Melting and boling point: K, Mass number: AMU

1 Code:

1.1 Description of code:

a) utils/corr.py — functions to find spearman and pearson correlation

b) utils/max_value.py — 2 functions that implement 2 algorithm to find maximum element in a list.

c) utils/sorting_.py — Implements bubble sort, merge sort, selection sort.

d) main.py — i) Samples all elements and related properties ii) Time complexity plots

1.2 How to run the code:

Run main.py — generates output of all the questions and plot as well.

2 Select any 20 elements from the periodic table. Obtain the atomic number, mass, density, melting point, and boiling point.

code: function make_property_list in main.py

Atomic Numbers Sampled
[15, 4, 36, 32, 29, 18, 14, 70, 12, 55, 5, 80, 73, 28, 30, 65, 74, 26, 46, 42]
Mass Numbers Sampled
[30.974, 9.0122, 83.798, 72.63, 63.546, 39.95, 28.085, 173.05, 24.305, 132.91, 10.81, 200.59, 180.95, 58.693, 65.38, 158.93, 183.84, 55.845, 106.42, 95.95]
Densities Sampled
[1.823, 1.85, 0.003749, 5.323, 8.96, 0.001784, 2.329, 6.9, 1.738, 1.93, 2.34, 13.534, 16.69, 8.908, 7.14, 8.23, 19.25, 7.874, 12.023, 10.28]
Melting Point Sampled
[317.3, 1560, 115.78, 1211.4, 1357.77, 83.81, 1414, 1097, 923, 301.7, 2349, 234.321, 3290, 1728, 692.68, 1629, 3695, 1811, 1828.05, 2896]
Boiling Point Sampled
[553.7, 2742, 119.93, 3106, 2835, 87.302, 3538, 1469, 1363, 944, 4200, 629.88, 5731, 3003, 1180, 3396, 6203, 3134, 3236, 4912]

2.1 Write two algorithms to find the elements having maximum value of each of the properties.

code - utils/max_value.py

Algorithm 1 Max Value Algorithm 1

```
1: procedure MaxValueAlgorithm1(lis)
2:     largest ← lis[0]
3:     for i ← 1 to length(lis) do
4:         if lis[i] > largest then
5:             largest ← lis[i]
6:         end if
7:     end for
8:     return largest
9: end procedure
```

Algorithm 2 Max Value Algorithm 2

```
1: procedure MaxValueAlgorithm2(lis)
2:     for i ← 0 to length(lis) do
3:         for j ← 0 to length(lis) − i − 1 do
4:             if lis[j] > lis[j + 1] then
5:                 swap lis[j] and lis[j + 1]
6:             end if
7:         end for
8:     end for
9:     return lis[−1]
10: end procedure
```

	Algo 1	**Algo 2**
Maximum Atomic Number	80	80
Maximum Mass Number	200.59	200.59
Maximum Density	19.25	19.25
Maximum Melting Point	3695	3695
Maximum Boiling Point	6203	6203

2.2 Sort the atoms in the increasing order of melting point, and boiling point using at least three different sorting algorithms.

code - utils/sorting_.py

Algorithm	**Best Case**	**Average Case**	**Worst Case**
bubble sort	O(n)	$O(n^2)$	$O(n^2)$
merge sort	O(n log n)	O(n log n)	O(n log n)
selection sort	$O(n^2)$	$O(n^2)$	$O(n^2)$

Bubble Sort Melting Point
[83.81, 115.78, 234.321, 301.7, 317.3, 692.68, 923, 1097, 1211.4, 1357.77, 1414, 1560, 1629, 1728, 1811, 1828.05, 2349, 2896, 3290, 3695]
Merge Sort Melting Point
[83.81, 115.78, 234.321, 301.7, 317.3, 692.68, 923, 1097, 1211.4, 1357.77, 1414, 1560, 1629, 1728, 1811, 1828.05, 2349, 2896, 3290, 3695]
Selection Sort Melting Point
[83.81, 115.78, 234.321, 301.7, 317.3, 692.68, 923, 1097, 1211.4, 1357.77, 1414, 1560, 1629, 1728, 1811, 1828.05, 2349, 2896, 3290, 3695]
Bubble Sort Boiling Point
[87.302, 119.93, 553.7, 629.88, 944, 1180, 1363, 1469, 2742, 2835, 3003, 3106, 3134, 3236, 3396, 3538, 4200, 4912, 5731, 6203]
Merge Sort Boiling Point
[87.302, 119.93, 553.7, 629.88, 944, 1180, 1363, 1469, 2742, 2835, 3003, 3106, 3134, 3236, 3396, 3538, 4200, 4912, 5731, 6203]
Selection Sort Boiling Point
[87.302, 119.93, 553.7, 629.88, 944, 1180, 1363, 1469, 2742, 2835, 3003, 3106, 3134, 3236, 3396, 3538, 4200, 4912, 5731, 6203]

3 Compute the correlation between:

code - utils/corr.py

$$Pearson\_correlation = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

where, x_i and y_i are the values of the two variables for the i-th data point, $\bar{x}$ and $\bar{y}$ are the means of the two variables, and n is the number of data points.

$$Spearman\_correlation = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{2}$$

where, d_i is the difference between the rank of X and Y for the i-th value in the dataset and n is the number of data points.

Property	Pearson Correlation	Spearman Correlatio
mass and density	0.7187010914596609	0.3203007518796992
melting pt and boiling pt	0.9765606629531	0.6691729323308271 814
density and melting point	0.6767554039076843	0.43157894736842106

Inference: We see maximum correlation between Melting pt and boiling pt. Since both depend on type of bonding(covalent/ionic).

4 Time complexity plots: Carried out on Boiling point.

code - function time_complexity_analysis in main.py

Inference: The plot below agrees to table in section 2.2 wrt complexity of sorting algorithms. Finding maximum value through algo 2 is more complex than algo 1, because we first perform sorting in algo2 then choose the last element of the sorted list as the largest value.

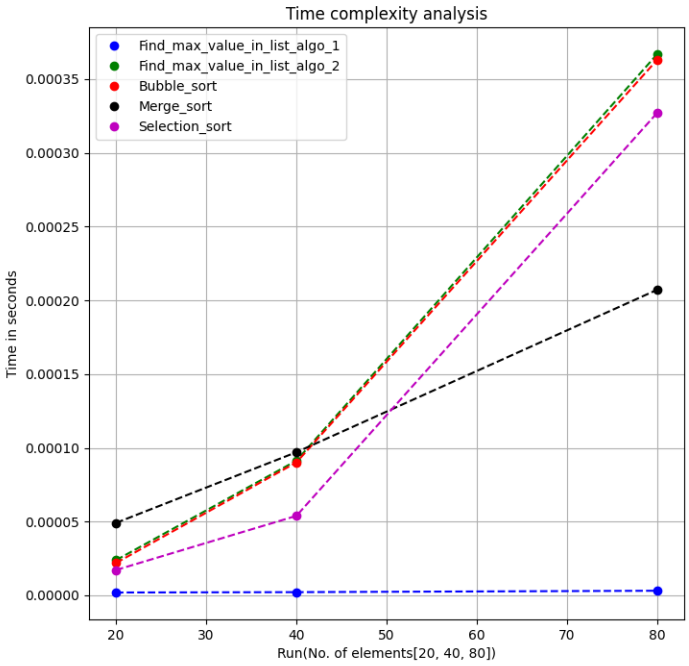

Figure 1: Time complexity analysis plot

5 References:

1. https://www.geeksforgeeks.org/
2. https://matplotlib.org/stable/tutorials/introductory/pyplot.htmls