

1. Selection Sort

9th.py - C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\9th.py (3.12.3)

File Edit Format Run Options Window Help

```
arr=[2,26,24,12]
n=len(arr)
for i in range(n-1):
    min=i
    for j in range(i+1,n):
        if arr[j]<arr[min]:
            min=j
    if min!=i:
        temp=arr[min]
        arr[min]=arr[i]
        arr[i]=temp
print(arr)
```

IDLE Shell 3.12.3

File Edit Shell Debug Options Window Help

Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

= RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\9th.py
Sorted array: [11, 12, 22, 25, 64]

>>>

== RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\9th.py ==
Sorted array: [2, 6, 15, 24, 26]
\

>>>

== RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\9th.py ==
[2, 12, 24, 26]

>>>

== RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\9th.py ==
[2, 12, 24, 26]


>>>

2. Bubble Sort

 bubble sortt.py - C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/bubble sortt.py (3.12.3)

File Edit Format Run Options Window Help

```
a=[2,5,6,7,3,12]
n=len(a)
for i in range(0,n-1):
    for j in range(i,n):
        if a[i]>a[j]:
            a[i],a[j]=a[j],a[i]
print(a)
```

 IDLE Shell 3.12.3

File Edit Shell Debug Options Window Help

Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>

= RESTART: C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/bubble sortt.py
[2, 3, 5, 6, 7, 12]

>>>

= RESTART: C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/bubble sortt.py
[2, 3, 5, 6, 7, 12]

>>>

3. Insertion sort

insertion.py - C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\insertion.py (3.12.3)

File Edit Format Run Options Window Help

```
n=[1,5,2,6,4]
a=len(n)
for i in range(0,a):
    key=n[i]
    j=i-1
    while j>0 and key<n[j]:
        n[j+1]=n[j]
        j-=1
    n[j+1]=key
print(n)
```

IDLE Shell 3.12.3

File Edit Shell Debug Options Window Help

Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

```
>>>
= RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\insertion.py
>>> [1, 2, 4, 5, 6]
```

```
===== RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\insertion.py =====
>>> [1, 2, 4, 5, 6]
```

4. Sequential Search

sequential.py - C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\sequential.py (3.12.3)

File Edit Format Run Options Window Help

```
arr = [5, 3, 7, 1, 9, 2]
target = 11
found = False
for index, element in enumerate(arr):
    if element == target:
        found = True
        break
if found:
    print(f"Element {target} found at index {index}.")
else:
    print(f"Element {target} not found in the list.")
```

IDLE Shell 3.12.3

File Edit Shell Debug Options Window Help

Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

```
>>>
= RESTART: C:\Users\Dharani M\AppData\Local\Programs\Python\Python312\sequential.py
>>> Element 11 not found in the list.
```

5.Brute-Force String Matching

seq in string.py - C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/seq in string.py (3.12.3)

```
File Edit Format Run Options Window Help
arr = ["apple", "banana", "strawberry", "papaya", "pomogranate", "grapes"]
target = "banana"
found = False
for index, element in enumerate(arr):
    if element == target:
        found = True
        break
if found:
    print(f"Element '{target}' {index}")
else:
    print(f"Element '{target}' not found ")
```

IDLE Shell 3.12.3

```
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/se
q in string.py
Element 'banana' 1
>>>
```

6.Closest-Pair

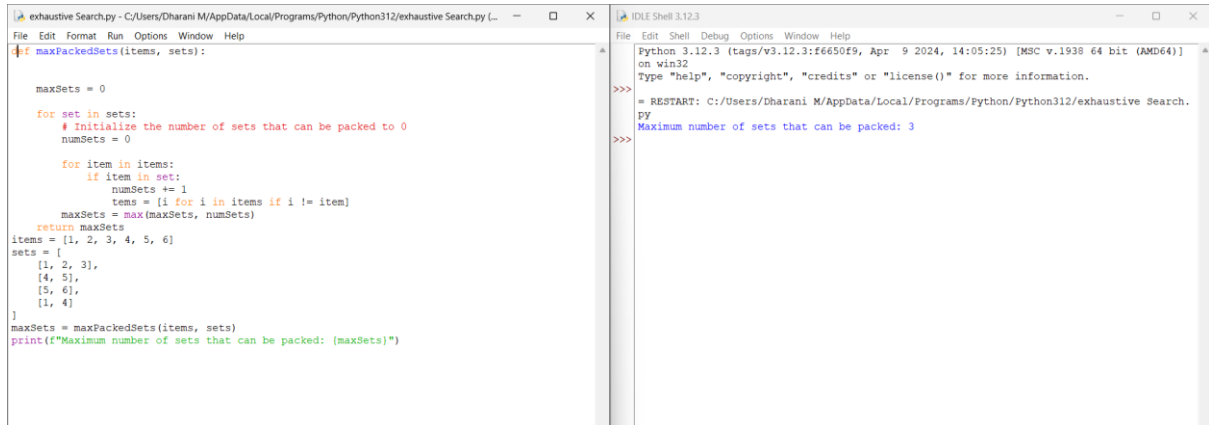
closets.py - C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/closets.py (3.12.3)

```
File Edit Format Run Options Window Help
import math
def closest_pair(points):
    min_dist = float('inf')
    closest = (None, None)
    for i in range(len(points)):
        for j in range(i + 1, len(points)):
            d = math.sqrt((points[i][0] - points[j][0])**2 + (points[i][1] - points[j][1])**2)
            if d < min_dist:
                min_dist = d
                closest = (points[i], points[j])
    return closest
points = [(2, 3), (5, 4), (6, 9), (7, 8)]
print(closest_pair(points))
```

IDLE Shell 3.12.3

```
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.
1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more informa
tion.
>>>
= RESTART: C:/Users/Dharani M/AppData/Local/Programs/Python/Python3
12/closets.py
((6, 9), (7, 8))
>>>
= RESTART: C:/Users/Dharani M/AppData/Local/Programs/Python/Python3
12/closets.py
((6, 9), (7, 8))
>>>
```

8.Exhaustive Search



The image shows a Python IDE with two windows. The left window, titled 'exhaustive Search.py', contains a function `maxPackedSets` that iterates through a list of items and sets to find the maximum number of sets that can be packed. The right window, titled 'IDLE Shell 3.12.3', shows the output of the function: 'Maximum number of sets that can be packed: 3'.

```
exhaustive Search.py - C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/exhaustive Search.py L
File Edit Format Run Options Window Help
def maxPackedSets(items, sets):
    maxSets = 0
    for set in sets:
        # Initialize the number of sets that can be packed to 0
        numSets = 0
        for item in items:
            if item in set:
                numSets += 1
                items = [i for i in items if i != item]
        maxSets = max(maxSets, numSets)
    return maxSets
items = [1, 2, 3, 4, 5, 6]
sets = [
    [1, 2, 3],
    [4, 5],
    [5, 6],
    [1, 4]
]
maxSets = maxPackedSets(items, sets)
print(f"Maximum number of sets that can be packed: {maxSets}")

IDLE Shell 3.12.3
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [MSC v.1938 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Dharani M/AppData/Local/Programs/Python/Python312/exhaustive Search.
PY
Maximum number of sets that can be packed: 3
>>>
```