

Task-5: Implement various searching and sorting operations in Python

Aim: To implement various searching and sorting operations in python

5.A) A company stores employee records in a list of Dictionaries, where each dictionary contains id, name and dept. write a func to find emp by id that takes this list and a target Employee ID as arg and returns the Dictionary of the emp

Algorithm:

1. Input def
2. Define the function
 - i) a list of dictionaries, where each represents an employee record with key id, name and dept
 - ii) An int representing emp id
3. Iterate the list
4. check for matching ID
5. Return Matching Record:
 - i) if match is found, return the Dictionary
6. Handle no match
if the loop completes without finding a match, return none.

Program

```
def find_employee_by_id (employees, target_id):
```

```
    for employee in employees:
```

```
        if employee[id] == target_id:
```

```
            return employee
```

```
    return None
```

```
return None
```

```
# test the function
```

```
employees = [
```

```
    {
```

```
        'id': 1, 'name': 'Alice', 'dept': 'HR',
```

```
        'id': 2, 'name': 'Bob', 'dept': 'Engineering',
```

```
        'id': 3, 'name': 'Charlie', 'dept': 'Sales',
```

```
    ]
```

```
print(find_employee_by_id(employees, 2))
```

5.2 You are developing grade management system for a school. The system maintains a list of students records. where each record is represent as a report that displays students Score ascending order.

Algorithm:

1. initialization:

. Get the length of the students

2. Outer loop:

iterate from $i = 0$ to $n-1$. This loop represents the no. of passes.

3. track swaps: initialize a boolean variable swapped to false.

output :-

```
{ 'id': 2, 'name': 'bob', 'dept': 'Engineering' }
```

4. inner loop:

- iterate from $j=0$ to $n-2$. This loop compares adjacent elements in the list.

5. compare and swap:

- For each pair of adjacent elements
 - i) compare their score values
 - ii) if $\text{students}[i][\text{score}] > \text{students}[i+1][\text{score}]$
 - iii) set swapped true to indicate that a swap was made.

6. Early termination

- After each pass of the inner loop, check if swapped is false. If swapped is false, if no swaps were made during the pass, the list is already sorted.

7. completion:

The function modifies the stu list in place.

Output:

Before sorting:

{ 'name': 'Alice', 'score': 88 }

{ 'name': 'Bob', 'score': 95 }

{ 'name': 'Charlie', 'score': 75 }

{ 'name': 'Diana', 'score': 85 }

After sorting:

{ 'name': 'Charlie', 'score': 75 }


{ 'name': 'Diana', 'score': 85 }

{ 'name': 'Alice', 'score': 88 }

{ 'name': 'Bob', 'score': 95 }

Program

```
def bubble_sort_scores(students):  
    n = len(students)  
    for i in range(n):  
        swapped = False  
        for j in range(0, n-i-1):  
            if students[j]['score'] > students[j+1]['score']:  
                students[j], student[j+1] = students[j+1], students[j]  
                swapped = True  
        if not swapped:  
            break  
  
students = [  
    {'name': 'Alice', 'score': 88},  
    {'name': 'Bob', 'score': 95},  
    {'name': 'Charlie', 'score': 75},  
    {'name': 'Diana', 'score': 85}  
]  
  
Print('Before sorting:')  
for student in students:  
    Print(student)  
  
bubble_sort_scores(students)  
  
Print("\nAfter sorting:")  
for student in students:  
    Print(student).
```



Program

```
def bubble_sort_scores(students):  
    n = len(students)  
    for i in range(n):  
        swapped = False  
        for j in range(0, n-i-1):  
            if students[j]['score'] > students[j+1]['score']:  
                students[j], students[j+1] = students[j+1], students[j]  
                swapped = True  
        if not swapped:  
            break  
students = [  
    {'name': 'Alice', 'score': 88},  
    {'name': 'Bob', 'score': 95},  
    {'name': 'Charlie', 'score': 75},  
    {'name': 'Diana', 'score': 85}  
]
```

```
Print("Before sorting")  
for student in students:  
    Print(student)  
bubble_sort_scores(students)  
Print("\n After sorting")  
for student in students:  
    Print(student)
```

VELTECH	
EX No.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
DATE	10/9

Result: Thus, various datatypes, List, tuples and Dictionary in python programming was used and verified successfully.