Experiment 1C

Shashwat Shah
60004220126
TY Btech Comps B

Aim: Perform Amortized Analysis using Potential method.

Theory: According to the computational complexity theory, the potential method is defined as: A method implemented to analyze the amortized time and space complexity of a data structure, a measure of its performance over sequence of operations that eliminate the cost of infrequent but expensive operations.

→ The potential approach focuses on how the current potient may be calculated directly from the algorithms or data structures present state.

→ The potential technique chooses a function ϕ that changes the data structure states into non-negative value.

→ At each stage of the computation, the potential function should be able to maintain the tracks of the precharged time.

→ It calculates the amount of time that can be saved up to cover expensive operations.

→ Intriguingly though it simply depends on the data structure current state, regardless of the history of the computation that led to that state.

→ We then define the amortized time of an operation as:

$$c + \phi(a') - \phi(a)$$

where c is the original cost of the operation and a and a' are the states of the data structure before

Scanned with CamScanner

and after the operation respectively.

→ As a result, the amortized time is calculated as the actual time plus the prospective change.

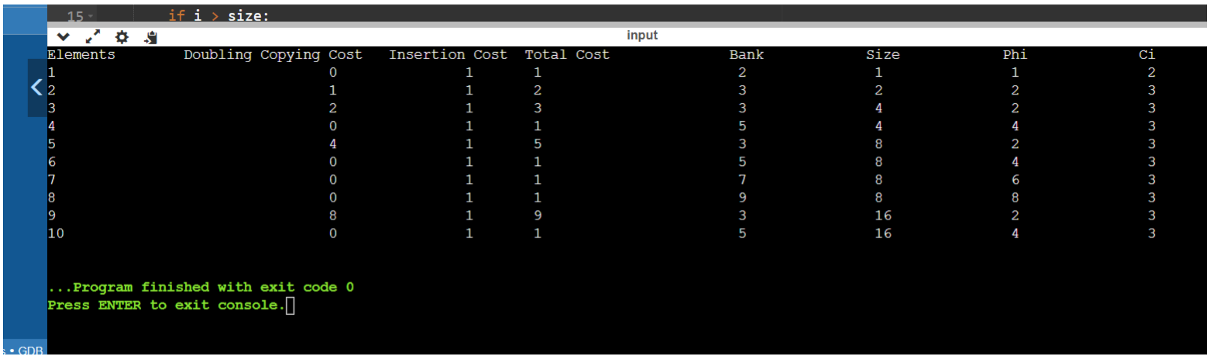
→ The amortized time of each operation should ideally be low when defined.

Conclusion: Hence, we studied the potential method

Scanned with CamScanner

```
def potential(n):
    size = 1
    total = 0
    dcost = 0
    icost = 0
    bank = 0
    phi = 0
    ci = 0
    phi_prev = 0
    print("Elements\tDoubling Copying Cost\tInsertion Cost\tTotal Cost\t\tBank\t\tSize\t\tPhi\t\tCi")
    for i in range(1, n + 1):
        icost = 1
        if i > size:
            size *= 2
            dcost = i - 1
        total = icost + dcost
        phi = 2 * i - size
        ci = total + phi - phi_prev
        bank += (3 - total)
        print(i, "\t\t\t\t", dcost, "\t\t", icost, "\t", total, "\t\t\t", bank, "\t\t", size, "\t\t", phi, "\t\t", ci)
        icost = 0
        dcost = 0
        phi_prev = phi
potential(10)
```

Output :



```
Elements        Doubling Copying Cost   Insertion Cost  Total Cost              Bank            Size            Phi             Ci
1                                0               1       1                       2               1               1               2
2                                1               1       2                       3               2               2               3
3                                2               1       3                       3               4               2               3
4                                0               1       1                       5               4               4               3
5                                4               1       5                       3               8               2               3
6                                0               1       1                       5               8               4               3
7                                0               1       1                       7               8               6               3
8                                0               1       1                       9               8               8               3
9                                8               1       9                       3               16              2               3
10                               0               1       1                       5               16              4               3


...Program finished with exit code 0
Press ENTER to exit console.
```