|  |  |
| --- | --- |
| Name: | Prerna Sunil Jadhav |
| Sap Id: | 60004220127 |
| Class: | T. Y. B. Tech (Computer Engineering) |
| Course: | Advance Algorithm Laboratory |
| Course Code: | DJ19CEL602 |
| Experiment No.: | 01-A |

**AIM: Perform Amortized Analysis of Multipop / Dynamic Tables / Binary Counter using**

**Aggregate, Accounting and Potential method. (Amortized Analysis)**

1A) Amortized Analysis (Aggregate method)

**CODE:**

class AggregateStack:

    def \_\_init\_\_(self):

        self.stack=[]

        self.cost=0

    def push(self,item):

        self.stack.append(item)

        self.cost+=1

        self.printstack()

        print("\tCost: ",self.cost)

    def pop(self):

        self.stack.pop()

        self.cost+=1

        self.printstack()

        print("\tCost: ",self.cost)

    def multipop(self,k):

        for i in range(k):

            self.pop()

    def printstack(self):

        print(self.stack,end='')

s=AggregateStack()

s.push(10)

s.push(10)

s.push(10)

s.push(10)

s.multipop(2)

print("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

def aggregate\_dynamic(n):

    size=1

    icost=0

    dcost=0

    totalcost=0

    total=0

    print("Element\tDoubling Cost\tInsertion cost\tTotal cost")

    for i in range(1,n+1):

        icost=1

        if i > size:

            size\*=2

            dcost=i-1

        totalcost=dcost+icost

        total=total+totalcost

        print(i,"\t\t",dcost,"\t\t",icost,"\t\t",totalcost,"")

        icost=0

        dcost=0

    return total/n

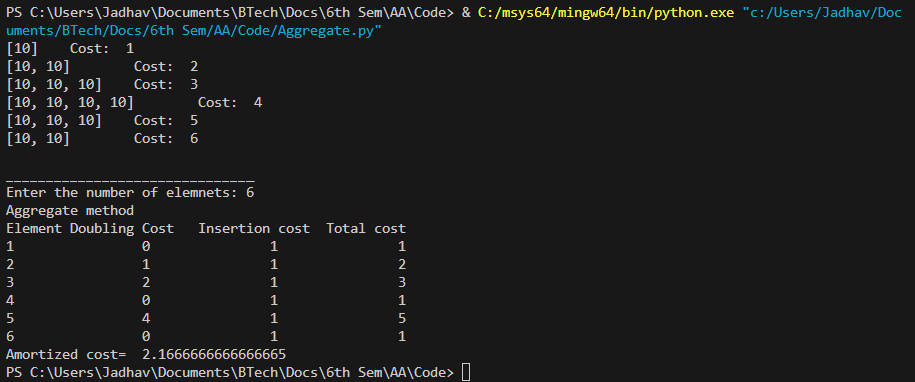
n=int(input("Enter the number of elemnets: "))

print("Aggregate method")

a=aggregate\_dynamic(n)

print("Amortized cost= ",a)

**OUTPUT:**

****