Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

Computer Engineering Department & Information Technology Engineering Department

Academic Year: 2021-2022

Class: S.Y.B.Tech **Sem.:** 4 **Course:** DAA

Name	**Pratik Pujari**		
UID no.	**2020300054**	**Class:**	**Comps C Batch**
Experiment No.	9		

AIM:	To implement approximation algorithms
THEORY:	**What is the Approximation algorithm?** An approximation algorithm is a way of dealing with NP-completeness for an optimization problem. This technique does not guarantee the best solution. The goal of the approximation algorithm is to come close as much as possible to the optimal solution in polynomial time. Such algorithms are called approximation algorithms or heuristic algorithms. **Features of Approximation Algorithm :** Here, we will discuss the features of the Approximation Algorithm as follows. • An approximation algorithm guarantees to run in polynomial time though it does not guarantee the most effective solution. • An approximation algorithm guarantees to seek out high accuracy and top quality solution(say within 1% of optimum) • Approximation algorithms are used to get an answer near the (optimal) solution of an optimization problem in polynomial time **Performance Ratios** Suppose we work on an optimization problem where every solution carries a cost. An Approximate Algorithm returns a legal solution, but the cost of that legal solution may not be optimal.

For Example, suppose we are considering for a minimum size vertex-cover (VC). An approximate algorithm returns a VC for us, but the size (cost) may not be minimized.

Another Example is we are considering for a maximum size Independent set (IS). An approximate Algorithm returns an IS for us, but the size (cost) may not be maximum. Let C be the cost of the solution returned by an approximate algorithm, and C* is the cost of the optimal solution.

We say the approximate algorithm has an approximate ratio P (n) for an input size n, where

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq P(n)$$

Intuitively, the approximation ratio measures how bad the approximate solution is distinguished with the optimal solution. A large (small) approximation ratio measures the solution is much worse than (more or less the same as) an optimal solution.

Applications of Approximation algorithm :

1. The Vertex Cover Problem

In the vertex cover problem, the optimization problem is to find the vertex cover with fewest vertices, and the approximation problem is to find the vertex cover with few vertices.

2. Travelling Salesman Problem

In the traveling salesperson problem, the optimization problem is to find the shortest cycle, and the approximation problem is to find a short cycle.

3. The Set Covering Problem

Computer Engineering Department &
Information Technology Engineering Department

Academic Year: 2021-2022

Class: S.Y.B.Tech **Sem.:** 4 **Course:** DAA

This is an optimization problem that models many problems that require resources to be allocated. Here, a logarithmic approximation ratio is used.

4. The Subset Sum Problem

In the Subset sum problem, the optimization problem is to find a subset of {x1,×2,×3…xn} whose sum is as large as possible but not larger than target value t.

Set Covering Problem

```
U = {1,2,3,4,5}
S = {S1,S2,S3}

S1 = {4,1,3},    Cost(S1) = 5
S2 = {2,5},      Cost(S2) = 10
S3 = {1,4,3,2},  Cost(S3) = 3

Output: Minimum cost of set cover is 13 and
        set cover is {S2, S3}

There are two possible set covers {S1, S2} with cost 15
and {S2, S3} with cost 13.
```

PSEUDOCODE:

Algorithm

1. $C \leftarrow 0$
2. While $C \neq U$ do

   Find the set whose cost effectiveness is smallest, say S

   Let $\alpha = \frac{c(S)}{|S - C|}$

   For each $e \in S\text{-}C$, set price(e) = α

   $C \leftarrow C \cup S$
3. Output picked sets

Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

Computer Engineering Department & Information Technology Engineering Department

Academic Year: 2021-2022

Class: S.Y.B.Tech **Sem.:** 4 **Course:** DAA

EXPERIMENT 1	
CODE:	import java.util.HashSet; import java.util.Scanner; import java.util.Set; public class set_covering { *// calculate minimum cost of set cover* public static void minCost(Set<Integer> Univ, Set<Integer> S[], int cost[], int m) { *// Set I is used to get minimum cost* Set<Integer> I = new HashSet<Integer>(); int minCost = 0; System.out.print("Best possible Solution sets\n"); *// Loop continues till the time I contains all the elements of U* while (!I.equals(Univ)) { *// calculate the min cost* double min = Double.MAX_VALUE; *// loop to find the minimum cost* int index = -1; double effCost[] = new double[m]; for (int i = 0; i < m; i++) { *// diff is difference between S[i] and I* Set<Integer> diff = new HashSet<Integer>(S[i]); diff.removeAll(I); *// if size of diff != 0* if (diff.size() != 0) { effCost[i] = cost[i] / diff.size(); } *// if size of diff = 0* else { effCost[i] = Double.MAX_VALUE; } *// change min if effCost[i] < min* if (min > effCost[i]) {

```
                min = effCost[i];
                index = i;
            }
        }
        // Printing the S[i]
        System.out.print("S[" + (index + 1) + "] ");
        // Used to get union of I and S[index]
        Set<Integer> union = new HashSet<Integer>(I);
        union.addAll(S[index]);
        // I = union of I and S[index]
        I = union;
        // Min cost
        minCost = minCost + cost[index];
    }
    System.out.println(" Minimum cost: " + minCost);
  }

  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    // no of elements in U

    System.out.println("------------------SET COVERING
PROBLEM------------------\n");
    System.out.print("Enter the elements in Universal Set
: ");
    int n = sc.nextInt();
    // for storing the values in Universal set
    System.out.print("Enter the elements -> ");
    Set<Integer> Univ = new HashSet<Integer>();
    for (int i = 0; i < n; i++) {
      int temp = sc.nextInt();
      Univ.add(temp);
    }
    // no of sets in S
    System.out.print("Enter the number of sets : ");
    int m = sc.nextInt();
    // for storing the values in sets
    Set<Integer>[] S = new HashSet[m];
    int cost[] = new int[m];
```

Computer Engineering Department &
Information Technology Engineering Department

Academic Year: 2021-2022

Class: S.Y.B.Tech **Sem.:** 4 **Course:** DAA

```
        for (int i = 0; i < m; i++) {
            S[i] = new HashSet<Integer>();
            System.out.println("\nSet [" + (i + 1) + "] Details:
");
            // no of elements in S[i]
            System.out.print("Enter the elements : ");
            String input = sc.next();
            String[] elements = input.split(" ");
            for (int j = 0; j < elements.length; j++) {
                S[i].add(Integer.parseInt(elements[j]));
            }
            // Cost of S[i]
            System.out.print("Enter the cost(S[" + (i + 1) + "])
: ");
            cost[i] = sc.nextInt();
        }
        // method to get min cost
        minCost(Univ, S, cost, m);
        sc.close();
    }
}
```

OUTPUT:

```
------------ Subset Covering Prob ------------

Enter the elements in Universal Set : 5
Enter the elements -> 1 2 3 4 5
Enter the number of sets : 4

--------S[1]--------

Enter the elements of S[1]: 2
Enter the elements : 5 4
Enter the cost(S[1]) : 3

--------S[2]--------

Enter the elements of S[2]: 3
Enter the elements : 1 2 3
Enter the cost(S[2]) : 5
```

Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

Computer Engineering Department & Information Technology Engineering Department

Academic Year: 2021-2022

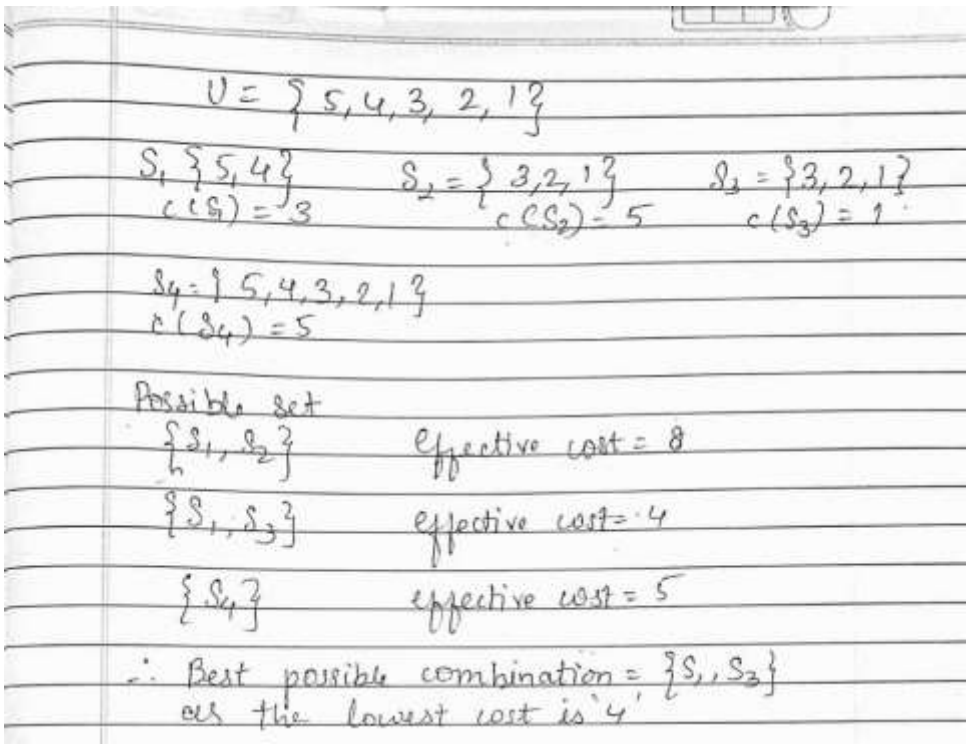
Class: S.Y.B.Tech **Sem.:** 4 **Course:** DAA

```
--------S[3]--------

Enter the elements of S[3]: 3
Enter the elements : 1 2 3
Enter the cost(S[3]) : 1

--------S[4]--------

Enter the elements of S[4]: 5
Enter the elements : 1 2 3 4 5
Enter the cost(S[4]) : 5
Best possible Solution sets is:
->S[3] S[1]  Minimum cost: 4
```

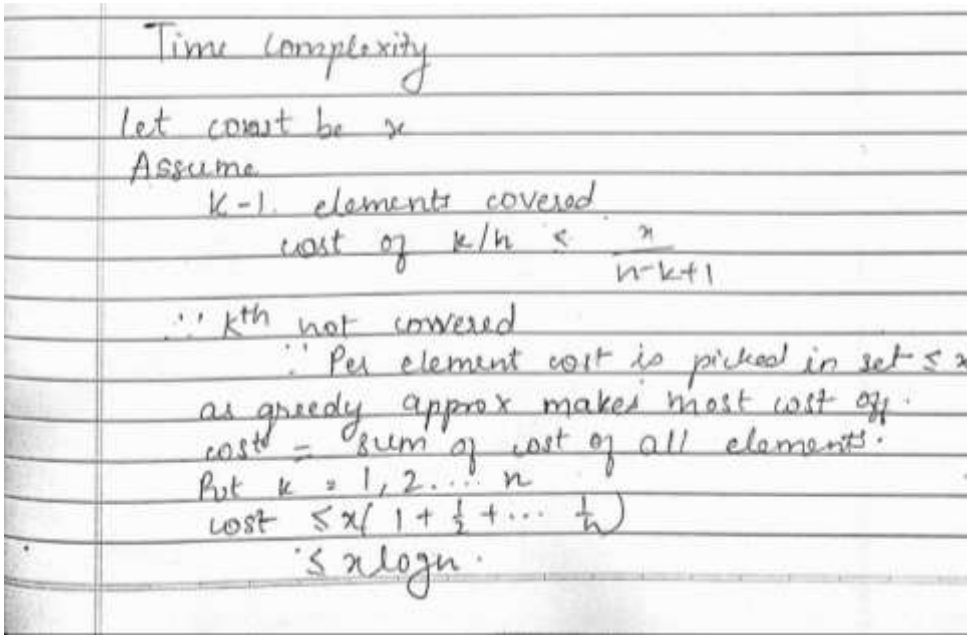


Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

Computer Engineering Department & Information Technology Engineering Department

Academic Year: 2021-2022

Class: S.Y.B.Tech **Sem.:** 4 **Course:** DAA

TIME COMPLEXITY	 **TIME & SPACE COMPLEXITY** The time complexity for approximation problem using greedy approximation approach is found out to be O(log n).

CONCLUSION: After the procedural programming through this experiment:

- Learnt about NP complete problem and how approximation algorithm is used to solve it.
- Implemented Set covering problem where used sets for storing the subsets and calculated the minimum cost for covering the sets.
- Derived the Time Complexity for the Set Covering Problem using Approximation algorithm and found it to be O(log n).
- Further for more explanation, I solved the Subset Covering Problem and for presented its working and commented on the time complexity which was derived and justified the derivation.