

## **Lab 3 Discrete**

### **Sets and Bit manipulation**

**Ahmed Ragy**

**22011690**

**John William**

**22010701**

## **Problem Statement**

There is a need for an efficient and scalable solution to perform essential set operations such as union, intersection, complement, difference, and cardinality. The implementation leverages bitwise operations for high-performance computations, representing sets as binary strings where each bit corresponds to an element in a universal set.

This solution aims to minimize computational overhead and provide an efficient tool for performing complex set operations.

## **Used Data Structures**

- ArrayList (List<Set> and List<String>):
- HashMap (Map<String, Integer>):
- String Array (String[])

## **Assumptions**

- It is assumed that the elements in the universal set are unique and distinct to avoid ambiguity in indexing and bitwise operations.
- All subsets provided by the user are assumed to be in the universal set.
- For operations involving two subsets (e.g., union, intersection, and difference), it is assumed the user selects valid subset indices.
- The implementation assumes case sensitivity for element names. For example, "Apple" and "apple" are treated as different elements.
- Subsets are referenced by integers preceded by the letter "S" (e.g., S1, S2). It is assumed that users will follow this naming convention when performing operations.
- The maximum size of the universal set is constrained by the size of the data type used for bitwise operations (e.g., 32 bits for integers).
- Subset names (e.g., S1, S2, ...) are assumed to be sequentially generated, starting from 1, to ensure consistent indexing.
- Error Handling for Invalid Operations.
- End Keyword (case-insensitive) terminate input for both the universal set
- The program assumes that the universal set is non-empty.

## Sample runs

### Test Case 1

```
\Users\ahmed\AppData\Local\Temp\cp_3quaw16f15dl
Enter Universe:
Enter "End" when you finish your entry.
ahmed
john
mohamed
ziad
end
Enter number of Sets:
2
Enter elements of Subset 1
ziad
john
mohamed
end
Enter elements of Subset 2
ahmed
john
end
U = [ahmed, john, mohamed, ziad]

Enter Operation: intersection
Enter 2 Sets to Intersect: 1 2
Intersection = [john]

Enter Operation: difference
Enter 2 Sets to difference: 1 2
Difference = [mohamed, ziad]

Enter Operation: complement
Enter Set to Complement: 2
Complement = [mohamed, ziad]
```

## Test Case 2

```
Enter End when you finish your entry.
```

```
10
```

```
20
```

```
30
```

```
40
```

```
50
```

```
end
```

```
Enter number of Sets:
```

```
3
```

```
Enter elements of Subset 1
```

```
10
```

```
20
```

```
end
```

```
Enter elements of Subset 2
```

```
30
```

```
end
```

```
Enter elements of Subset 3
```

```
end
```

```
U = [10, 20, 30, 40, 50]
```

```
Enter Operation: union
```

```
Enter 2 Sets to Union: 2 3
```

```
Union = [30]
```

```
Enter Operation: cardinality
```

```
Enter Set to get Cardinality: 3
```

```
Cardinality = 0
```

```
Enter Operation: intersection
```

```
Enter 2 Sets to Intersect: 1 3
```

```
Intersection = []
```

### Test Case 3

```
Enter Universe:
Enter "End" when you finish your entry.
element1
element2
end
Enter number of Sets:
1
Enter elements of Subset 1
element2
end
U = [element1, element2]

Enter Operation: difference
Invalid Input

Enter Operation: complement
Enter Set to Complement: 1
Complement = [element1]

Enter Operation: end
```

## Test Case 4

```
(Users\animesh\AppData\Local\Temp\cp_squaw18715u
Enter Universe:
Enter "End" when you finish your entry.
1
2
3
4
5
6
end
Enter number of Sets:
3
Enter elements of Subset 1
1
4
6
end
Enter elements of Subset 2
6
5
end
Enter elements of Subset 3
3
end
U = [1, 2, 3, 4, 5, 6]

Enter Operation: intersection
Enter 2 Sets to Intersect: 2 3
Intersection = []
```

## Test Case 5

```
Enter Universe:  
Enter "End" when you finish your entry.  
apple  
orange  
bannana  
peach  
pineapple  
end  
Enter number of Sets:  
2  
Enter elements of Subset 1  
apple  
bannana  
end  
Enter elements of Subset 2  
pineapple  
peach  
apple  
end  
U = [apple, orange, bannana, peach, pineapple]  
  
Enter Operation: union  
Enter 2 Sets to Union: 1 2  
Union = [apple, bannana, peach, pineapple]
```



## Test Case 6

```
(Users {ahmed\AppData\Local\Temp\ep_5qdw10r15ak5wark
Enter Universe:
Enter "End" when you finish your entry.
ahmed
john
hamada
end
Enter number of Sets:
1
Enter elements of Subset 1
hamada
john
end
U = [ahmed, john, hamada]

Enter Operation: complement
Enter Set to Complement: 1
Complement = [ahmed]

Enter Operation: cardinality
Enter Set to get Cardinality: 1
Cardinality = 2

Enter Operation: print
Enter Set to Print: 1
S1 = [hamada, john]
Binary Representation = 11

Enter Operation: end
```