SET DATA STRUCTURE - I

Introduction

- A set is a collection of objects need not to be in any particular order.
- It is just applying the mathematical concept in computer.
- Rules:
  - Elements should not be repeated.
  - Each element should have a reason to be in the set.

Set example :

➔ Assume we are going to store Indian cricketers in a set.

* The names should not be repeated, as well the name who is not in a team can't be inserted.

* This is the restriction we found in a set.

NULL SET(EMPTY SET)

- The set with no element is called null set or empty set.

For example:

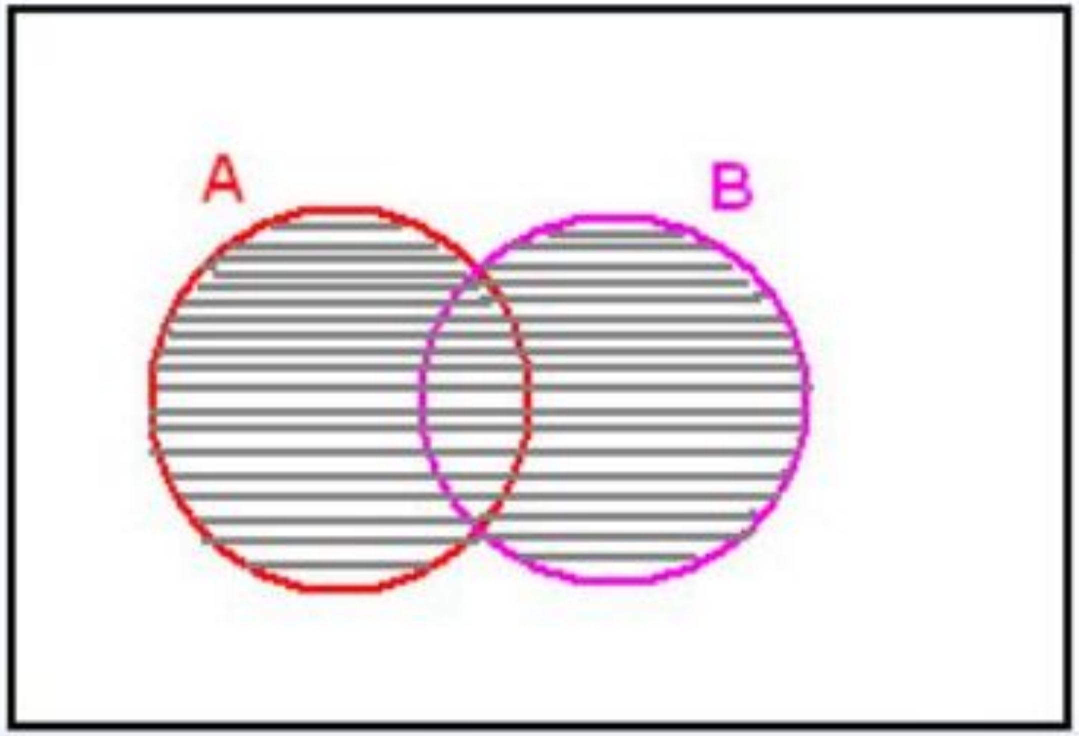
A={x | x is a prime number, where 24<x<29}

This set can’t contain a element. We don’t have any prime number within the limit 24 and 29.

A={ }

UNION

- Combine two or more sets(here two sets), without violating the rules for set.



Ex:

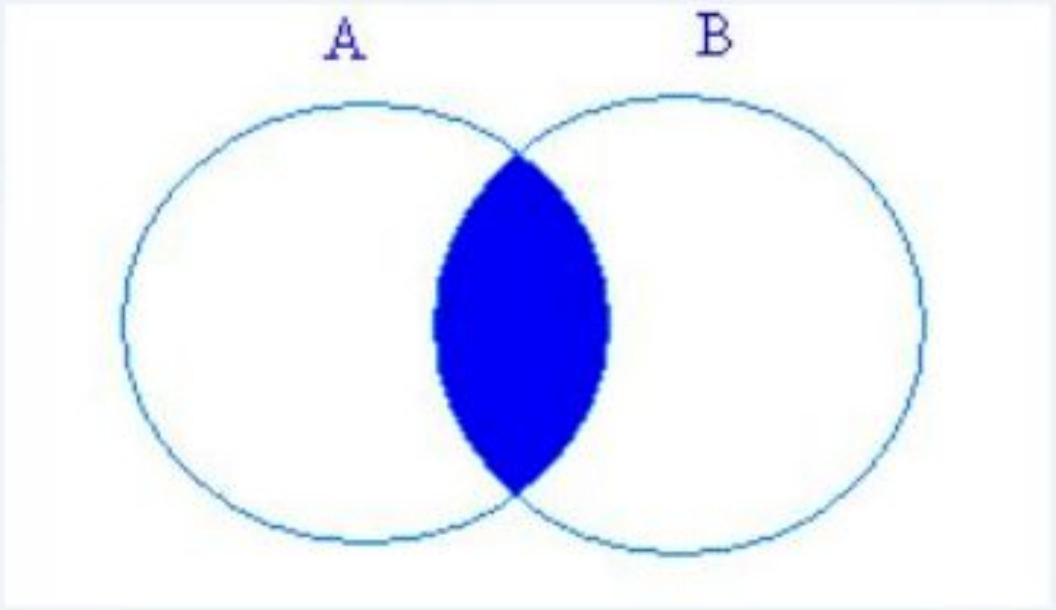
A={1,2,3,4,5}

B={3,4,5}

AUB={1,2,3,4,5}

INTERSECTION

- Gathering common elements in both the sets together as a single set.



Ex:

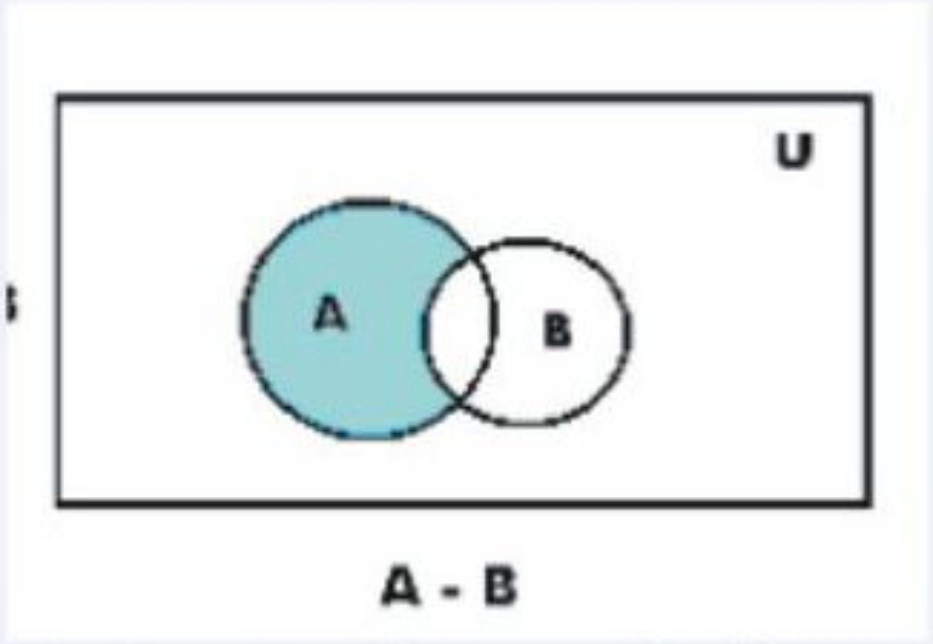
$A=\{1,2,3,4,5\}$

$B=\{3,4,5\}$

$A \cap B=\{3,4,5\}$

DIFFERENCE

- Forming a set with elements which are in first set and not in second set.



- A-B= {x| for all x belongs to A but not in B}

Ex:

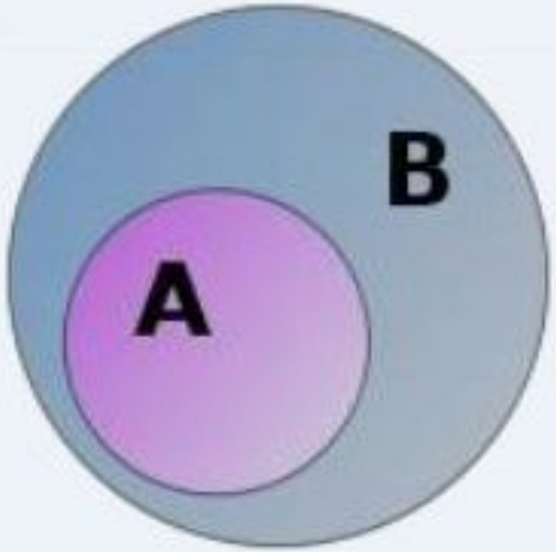
$A=\{1,2,3,4,5\}$ $B=\{3,4,5,6\}$

$A-B=\{1,2\}$

$B-A=\{6\}$

SUBSET

- ***int*** subset(*set1* , *set2*)
- Returns 1 if the set1 is the subset of the set2., otherwise 0 if not.
- Here, A c B(A contained in B)



Ex:

A={1,2}   B={1,2,3,4,5,6}

$A \subseteq B$

CARDINALITY

The number of elements in a set is the cardinality of that set.

The cardinality of the set A is often notated as $|A|$

Example

Let $A = \{1, 2, 3, 4, 5, 6\}$ and $B = \{2, 4, 6, 8\}$.

The cardinality of B is 4, since there are 4 elements in the set.

Equality of sets

Two sets are equal if and only if they have the same elements.

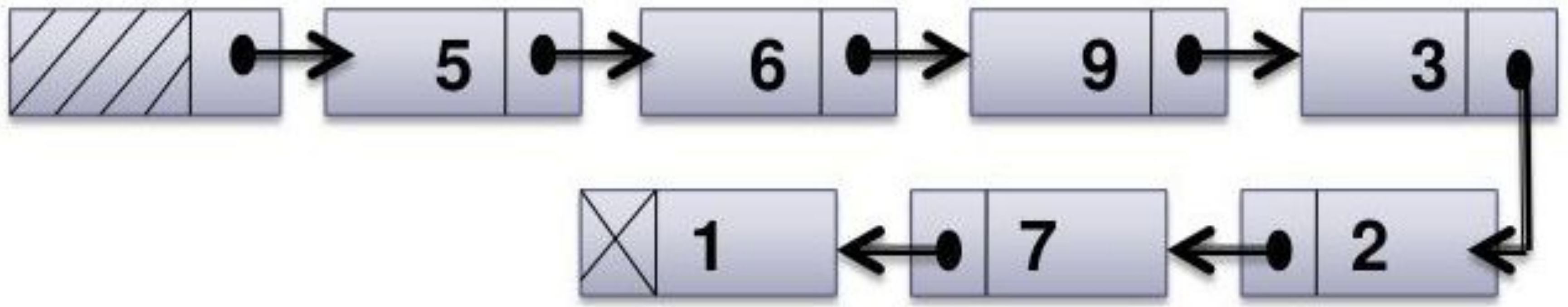
example **{*1, 2, 3*} = {*3, 2, 1*}** , that is the order of elements does not matte

Representation of Sets

- LIST
- HASH TABLE
- BIT VECTORS
- TREE

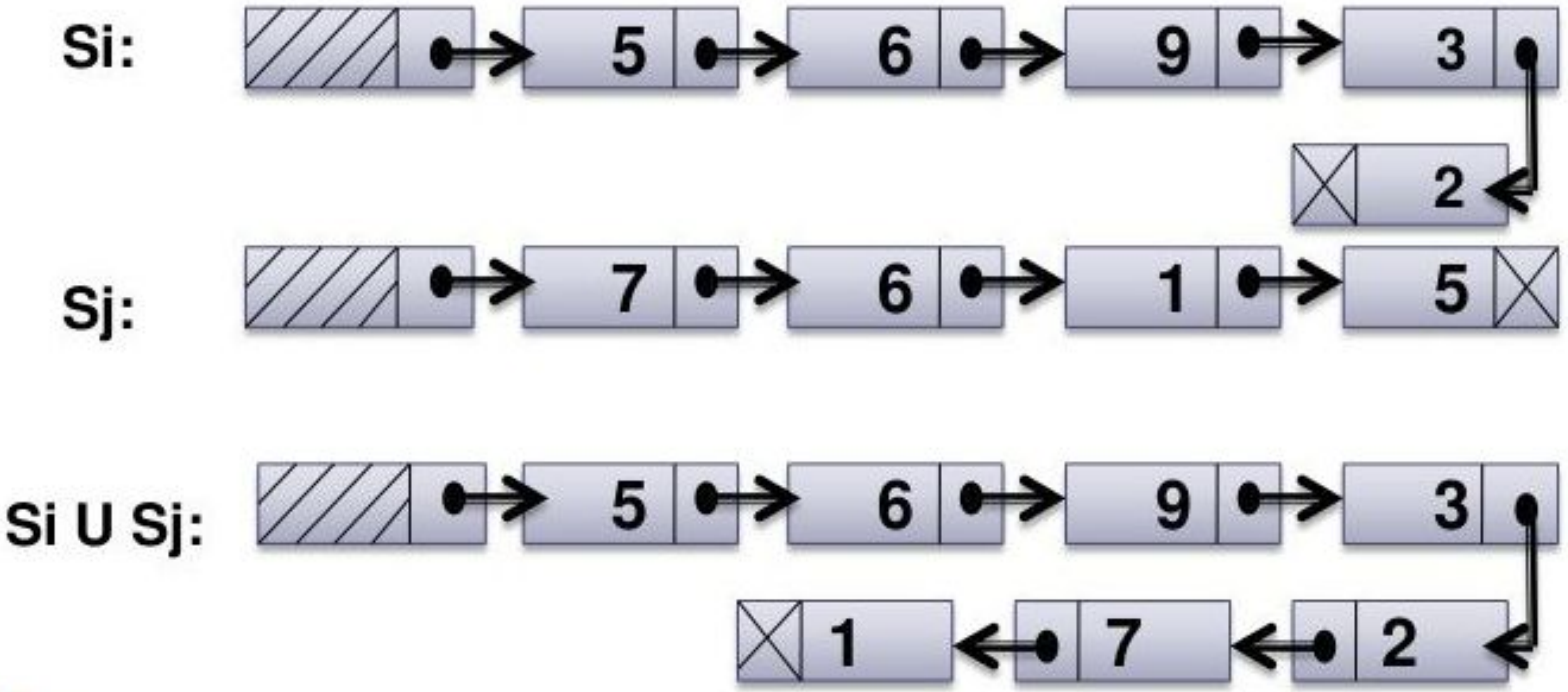
Linked List Trpresentation of Sets

- Simplest and straight forward
- Best suited for dynamic storage facility.
- This allow multiplicity of elements ie; Bag structure.
- All operations can be easily implemented and performance of these operations are as good as compared to other representations.
- Ex: set S = { 5,6,9,3,2,7,1} using linked list structure is

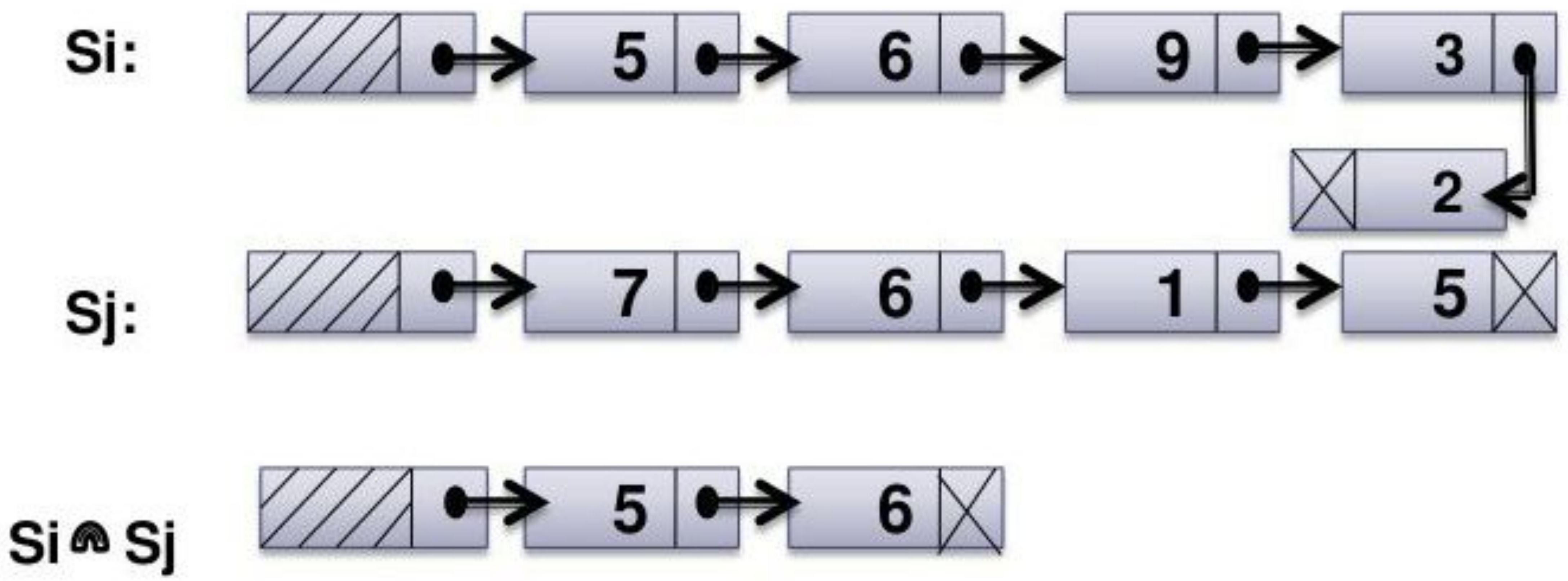


Operations on List Representation of sets

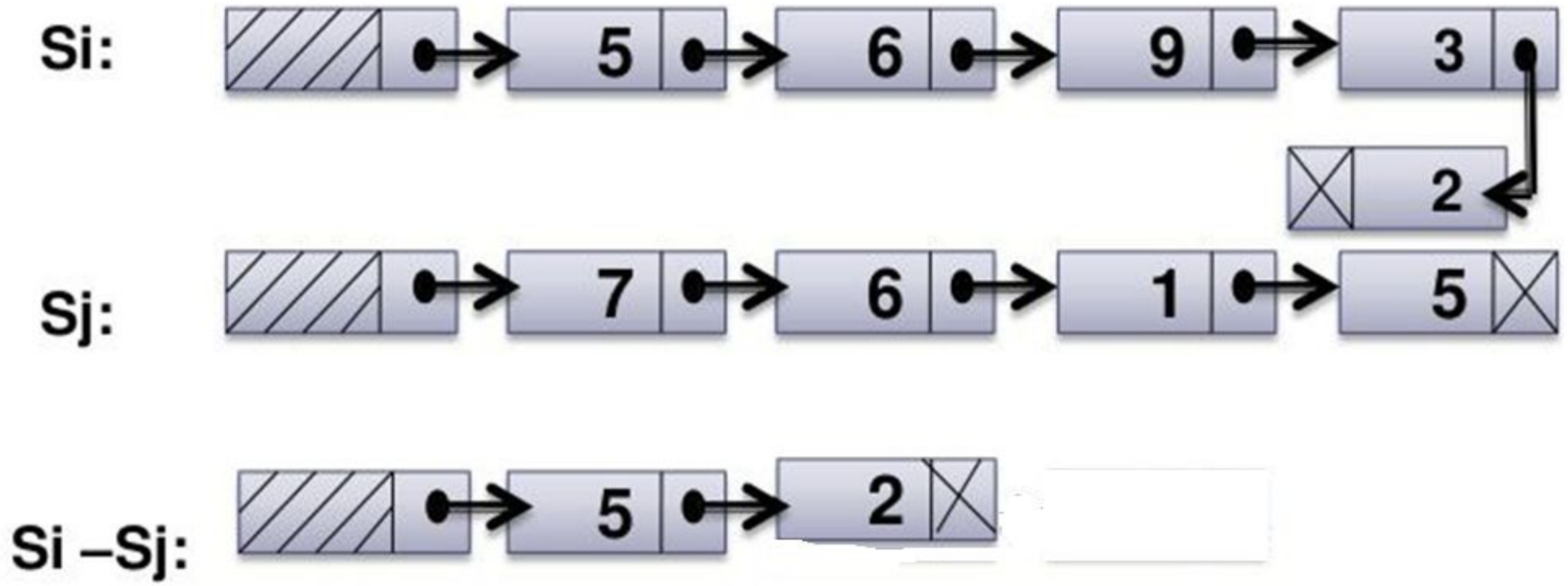
UNION:



INTERSECTION

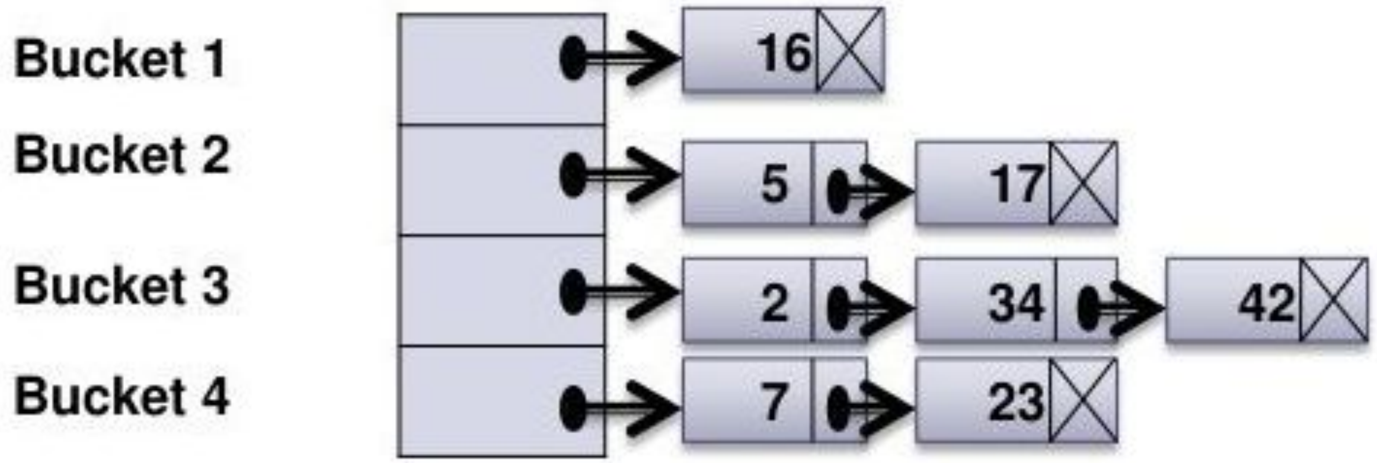


DIFFERENCE:

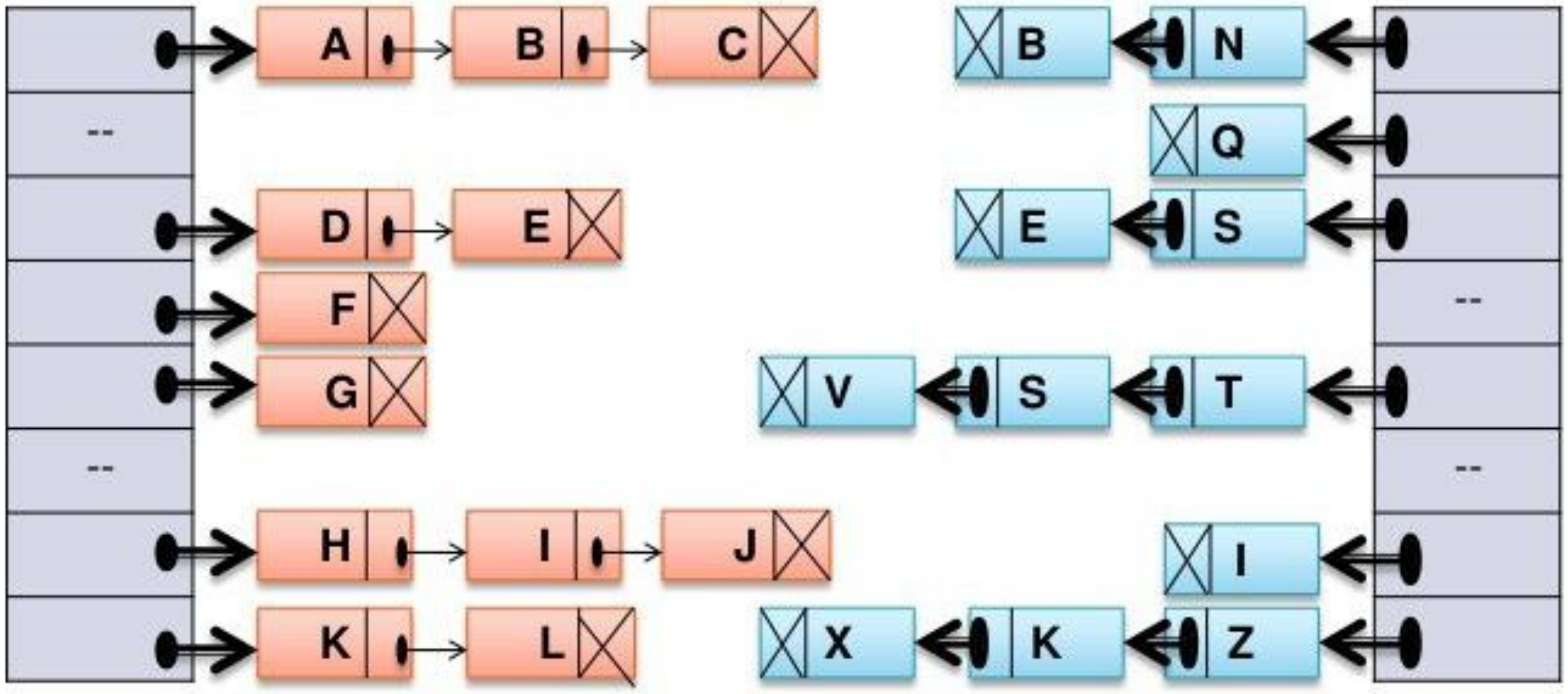


HASH TABLE REPRESENTATION OF SETS

- Here the elements in collection are separated in to number of buckets.
- Each bucket can hold arbitrary number of elements.
- Consider set S ={2,5,7,16,17,23,34,42}
- Here hash table with 4 buckets and H(x) hash function can store which can place element from S to any of the four buckets.



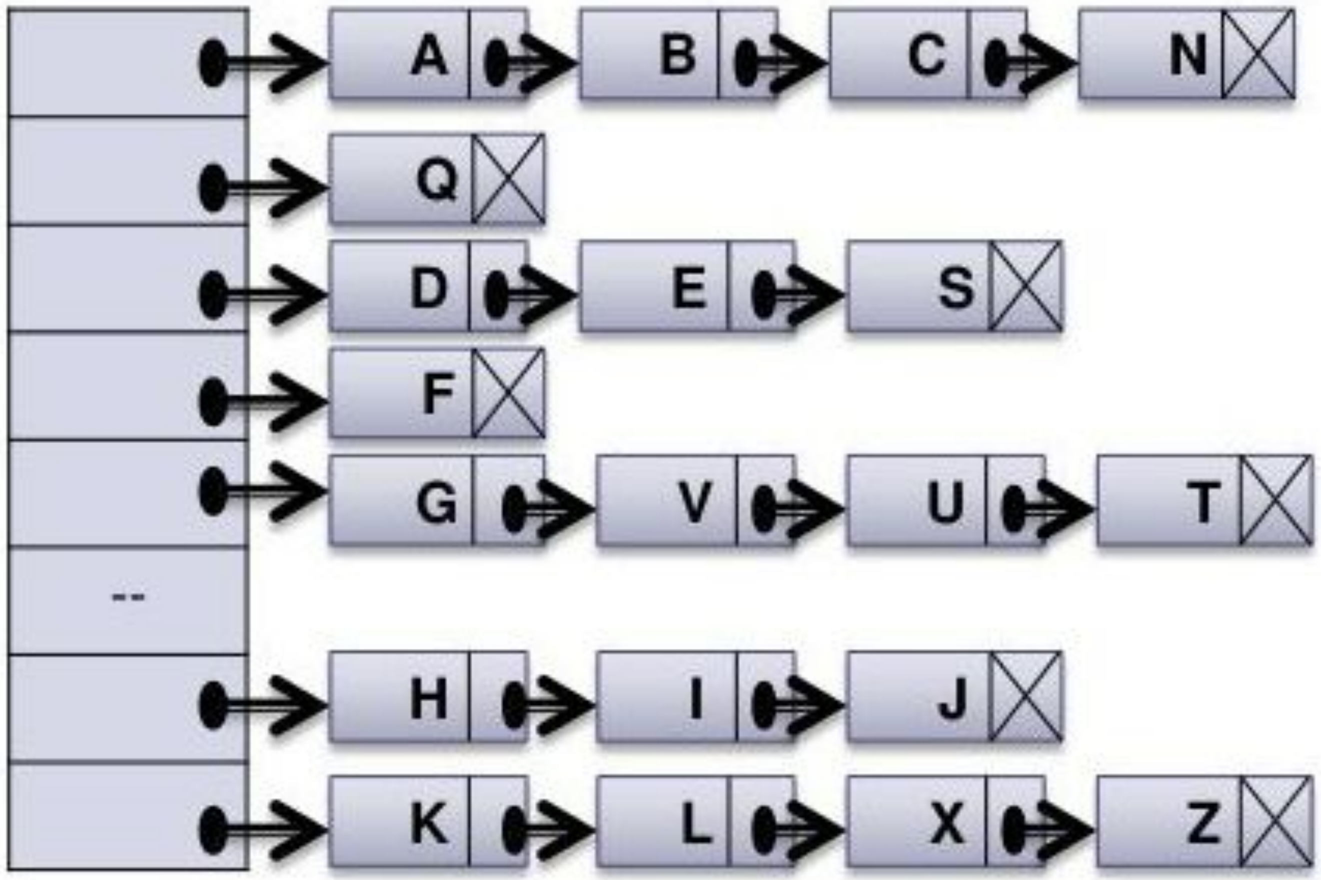
Operation on Hash table Representation of Sets



Set Si

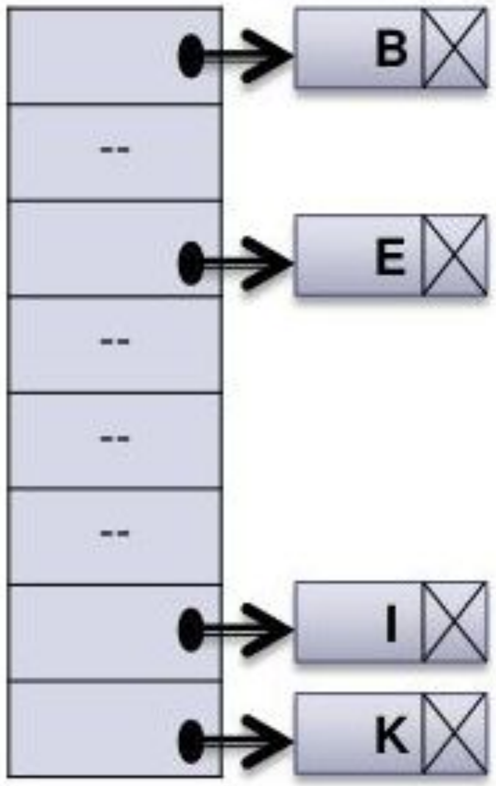
Set Sj

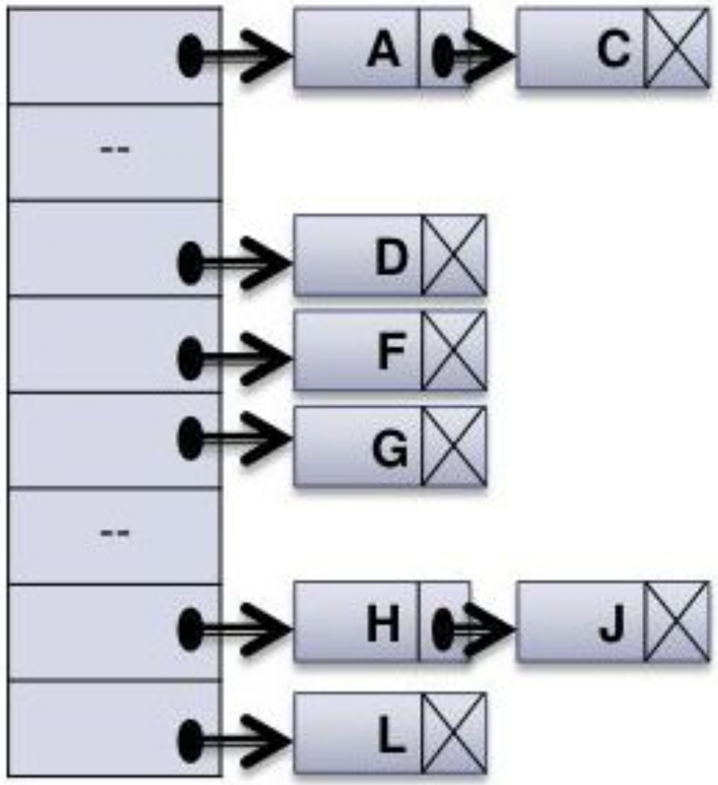
UNION: S = Si U Sj



INTERSECTION

DIFFERENCE



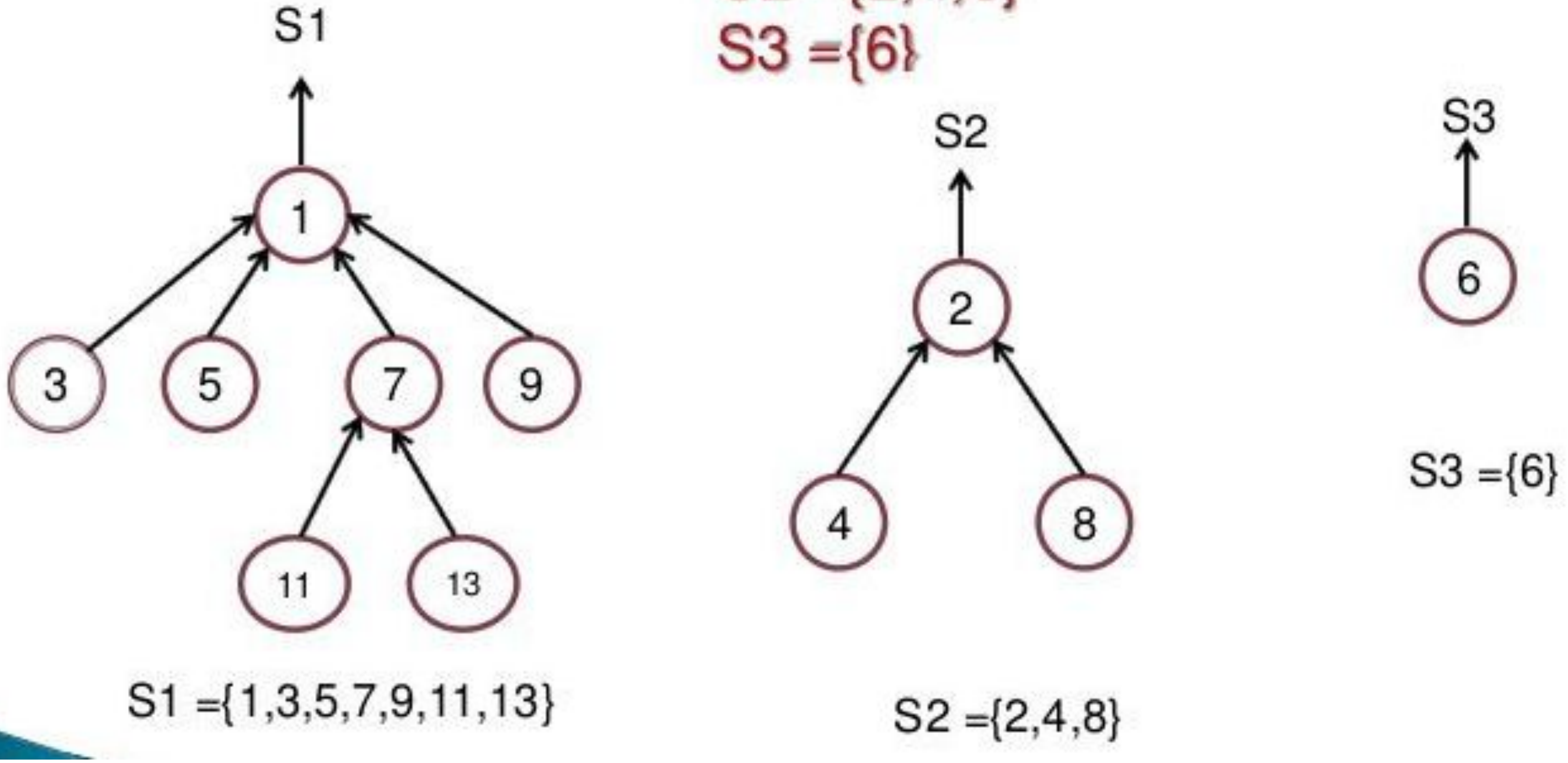


Si∩ Sj

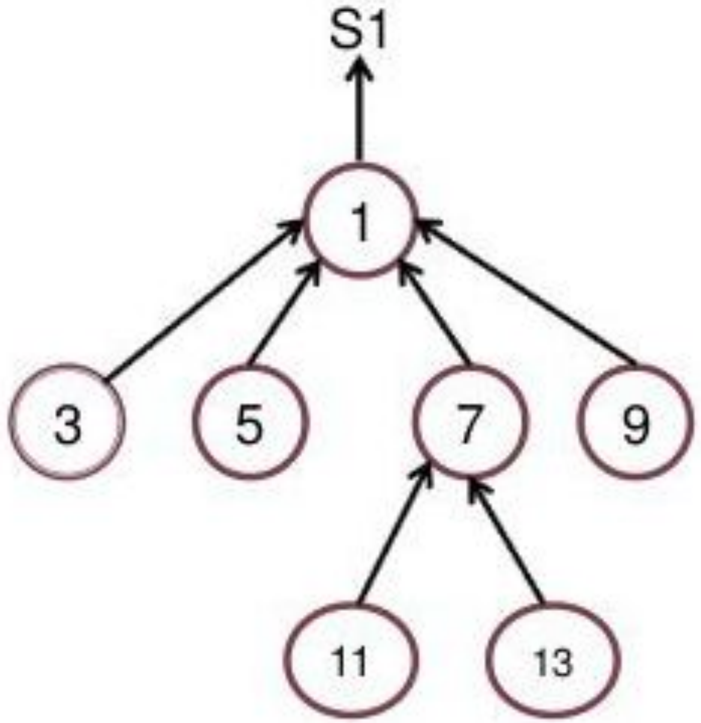
Si -Sj

TREE REPRESENTATION OF SETS

- Here a tree is used to represent one set, and the each element in the set has the same root.
- Each element in a set has pointer to its parent.
- Let us consider sets S1 ={1,3,5,7,9,11,13}
  S2 ={2,4,8}
  S3 ={6}

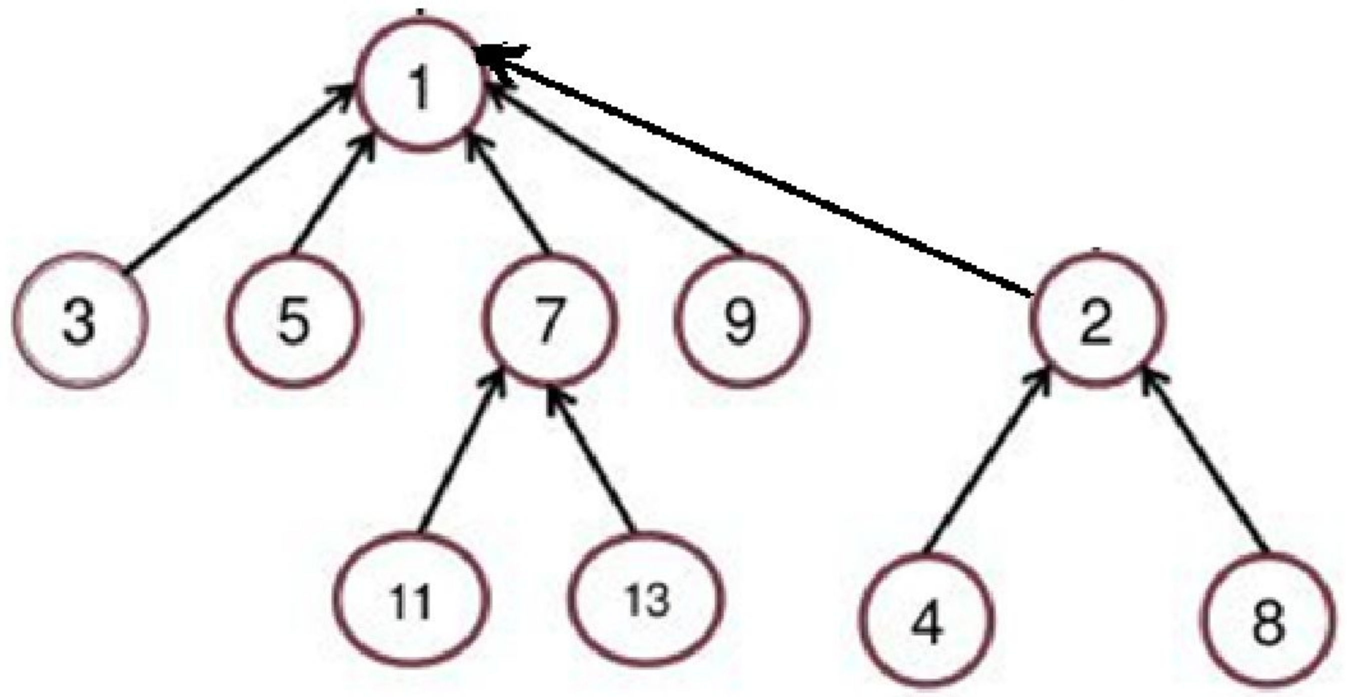


Tree representation of set S1 ={1,3,5,7,9,11,13}



0	--	1	--	1	--	1	--	1	--	7	--	7	--	--	--
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

S1 U S2



BIT VECTOR REPRESENTATION OF SETS

- **How to represent sets in the computer?**
- Assign a bit in a bit string to each element in the universal set and set the bit to 1 if the element is present otherwise use 0

Example:

All possible elements: U={1 2 3 4 5}

Assume A={2,5}

– Computer representation: A = 01001

Assume B={1,5}

– Computer representation: B = 10001

- A set, giving the records about the age of cricketer less than or equal to 35 is as given below:

  {0,0,0,0,1,1,1,1,0,1,1}
- Here 1 indicates the presence of records having the age less than or equal to 35.
- 0 indicates the absence of records having the age less than or equal to 35.
- As we have to indicate presence or absence of an element only, so 0 or 1 can be used for indication for saving storage space
- A bit array data structure is known for this purpose.
- A bit array is simply an array containing values 0 or 1(binary).

Operations on bit vector representation

- It is very easy to implement set operation on the bit array data structure.
- The operations are well defined only if the size of the bit arrays representing two sets under operation are of same size.

UNION

- To obtain the union of sets s_i and s_j, the bit-wise OR operation can be used
- S_i and S_j are given below:

$$
\begin{aligned}
S_i &= 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1 \\
S_j &= 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0 \\
S_i \cup S_j &= 1\ 0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1
\end{aligned}
$$

INTERSECTION

- To obtain the intersection of sets s_i and s_j, the bit-wise AND operation can be used
- S_i and S_j are given below:

$$
\begin{aligned}
S_i &= 1\,0\,0\,1\,0\,1\,1\,0\,0\,1 \\
S_j &= 0\,0\,1\,1\,1\,0\,0\,1\,0\,0 \\
S_i \cap S_j &= 0\,0\,0\,1\,0\,0\,0\,0\,0\,0
\end{aligned}
$$

DIFFERENCE

- The difference of S_i from S_j is the set of values that appear in S_i but not in $S_{j.}$ This can be obtained using bit-wise AND on the inverse of $S_{j.}$
- S_i and S_j are given below:

S_i = 1 0 0 1 0 1 1 0 0 1

S_j = 0 0 1 1 1 0 0 1 0 0

Sj' = 1 1 0 0 0 1 1 0 1 1

S = Si – Sj = Si ∩ Sj' =

1 0 0 0 0 1 1 0 0 1

Example 2:

Universal Set U= {1,2,3,4,5}

A={2,4,5}

Bit string corresponds to A={0,1,0,1,1}

B={ }

Bit string corresponds to B={0,0,0,0,0}

C=U

Bit string corresponds to C={1,1,1,1,1}

Universal Set U= {1,2,3,4,5}

A={2,4,5}
Bit string corresponds to A={0,1,0,1,1}

B={1,3,5 }
Bit string corresponds to B={1,0,1,0,1}

A∩B=5 {0,0,0,0,1)

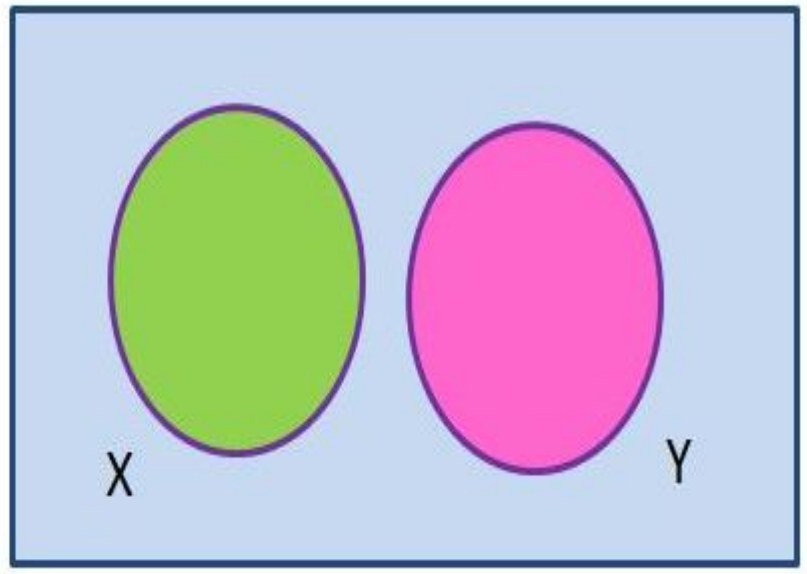
AUB={1,2,3,4,5}

A-b?
B-a?

Disjoint sets

A pair of sets which does not have any common element are called **disjoint sets**.

For example, set A={2,3} and set B={4,5} are disjoint sets.



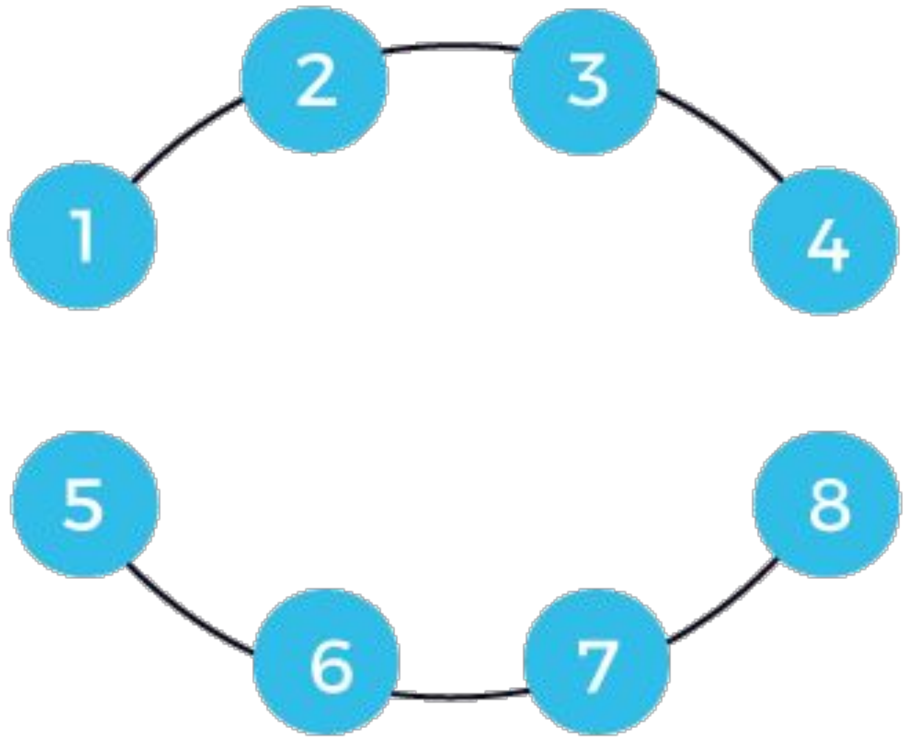
But set C={3,4,5} and {3,6,7} are not disjoint as both the sets C and D are having 3 as a common element.

Another definition: When the intersection of two sets is a null or empty set, then they are called disjoint sets.

Hence, if A and B are two disjoint sets, then;

$\mathbf{A \cap B = \varphi}$

s1 = {1, 2, 3, 4}
s2 = {5, 6, 7, 8}

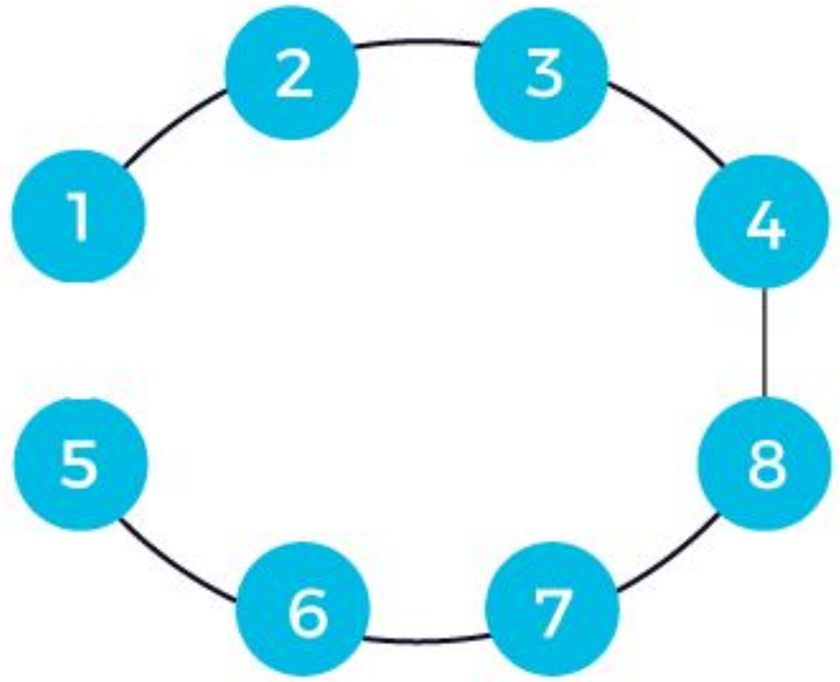


Since there is no common element between these two sets, we will not get anything if we consider the intersection between these two sets. This is also known as a disjoint set where no elements are common.

We can perform only two operations, i.e., find and union.

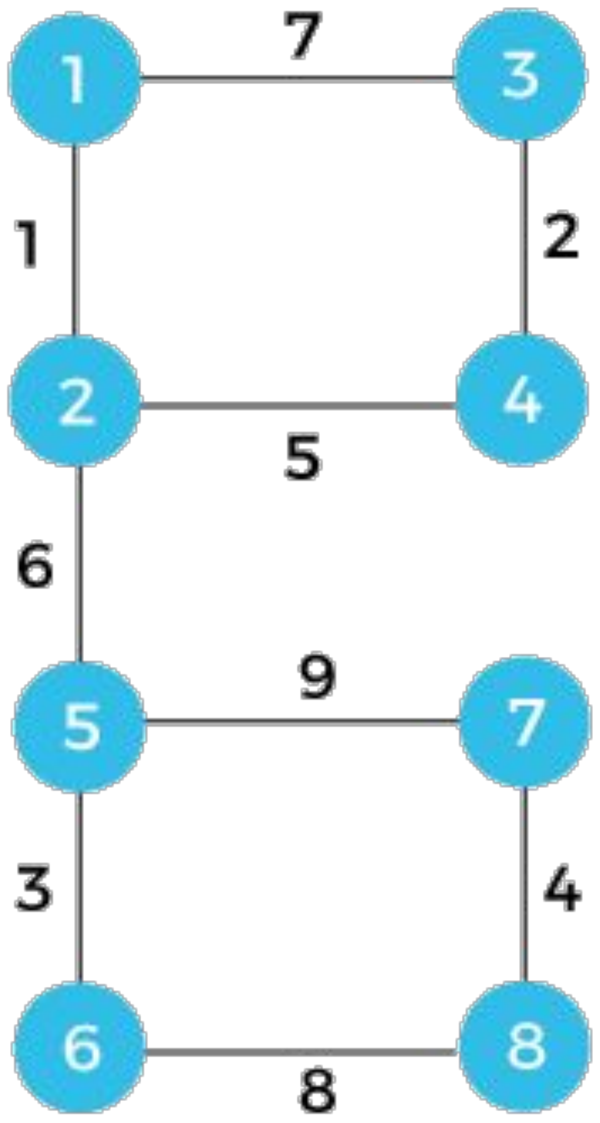
Suppose we want to perform the union operation on these two sets.
we have to check whether they are in different or same sets.
If they belong to the different sets, then we can perform the union operation;
otherwise, not.

For example, we want to perform the union operation between 4 and 8.
4 and 8 belong to different sets,
so we apply the union operation.



How can we detect a cycle in a graph?

We will understand this concept through an example. Consider the below example to detect a cycle with the help of using disjoint sets.



Tutorial :1

1.Bubble Sort

Data set: roll no,roll no*2,roll no-5,roll +10, roll no *3, roll no-10,roll no-15, roll no+8,roll no-20, rollno+20

2.Linearch

3.Binary search