ICOM 4035 – Data Structures

Dr. Manuel Rodríguez Martínez

Electrical and Computer Engineering Department

Lecture 6 – September 6th, 2001

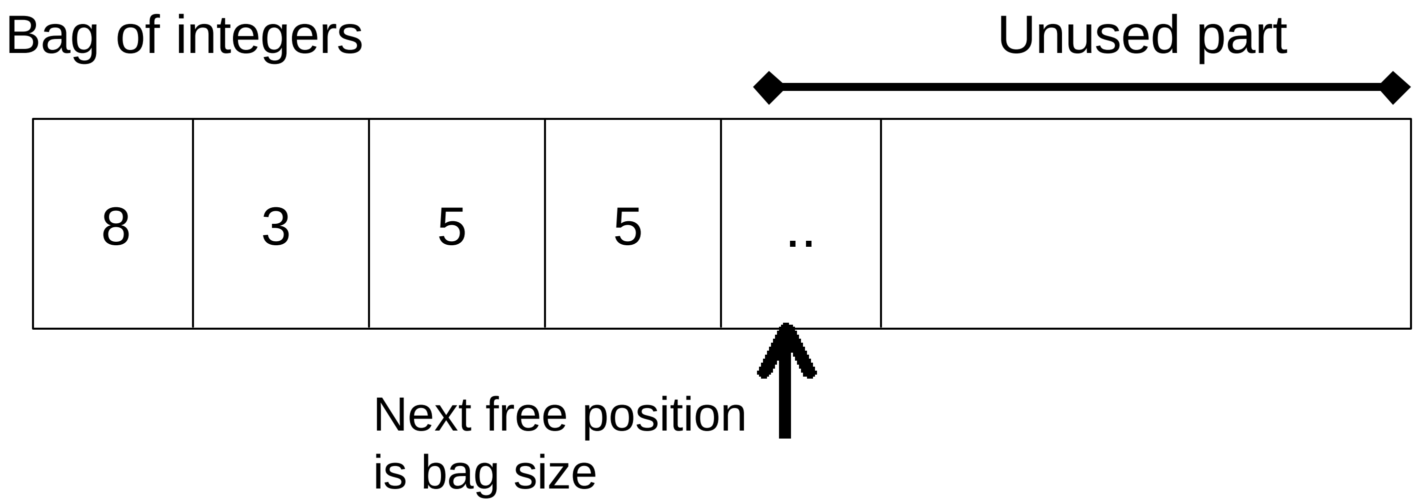
Readings

- Read Handout about Container Classes
  - Available from Engineering Reproduction Center as ICOM-4035-Manual # 2

Bag Class Container

- Bag (Multi-set) is data structure used to store elements with the following semantics:
  - Copies of the same element can stored in the bag B.
  - A find operation is supported to determine if an element x is present in the bag B.
  - A count operation is supported to determine the number of instances of element x in the bag B.
  - An erase operation is supported to erase an instance of an element x in the bag B.
  - An erase all operation is supported to erase all instances of an element x in the bag B
  - A union operation is supported to concatenate the contents of two bags.
- Bags are used to keep track of things in which copies are allowed
  - Movies in a video store, letters in a name, names is class room

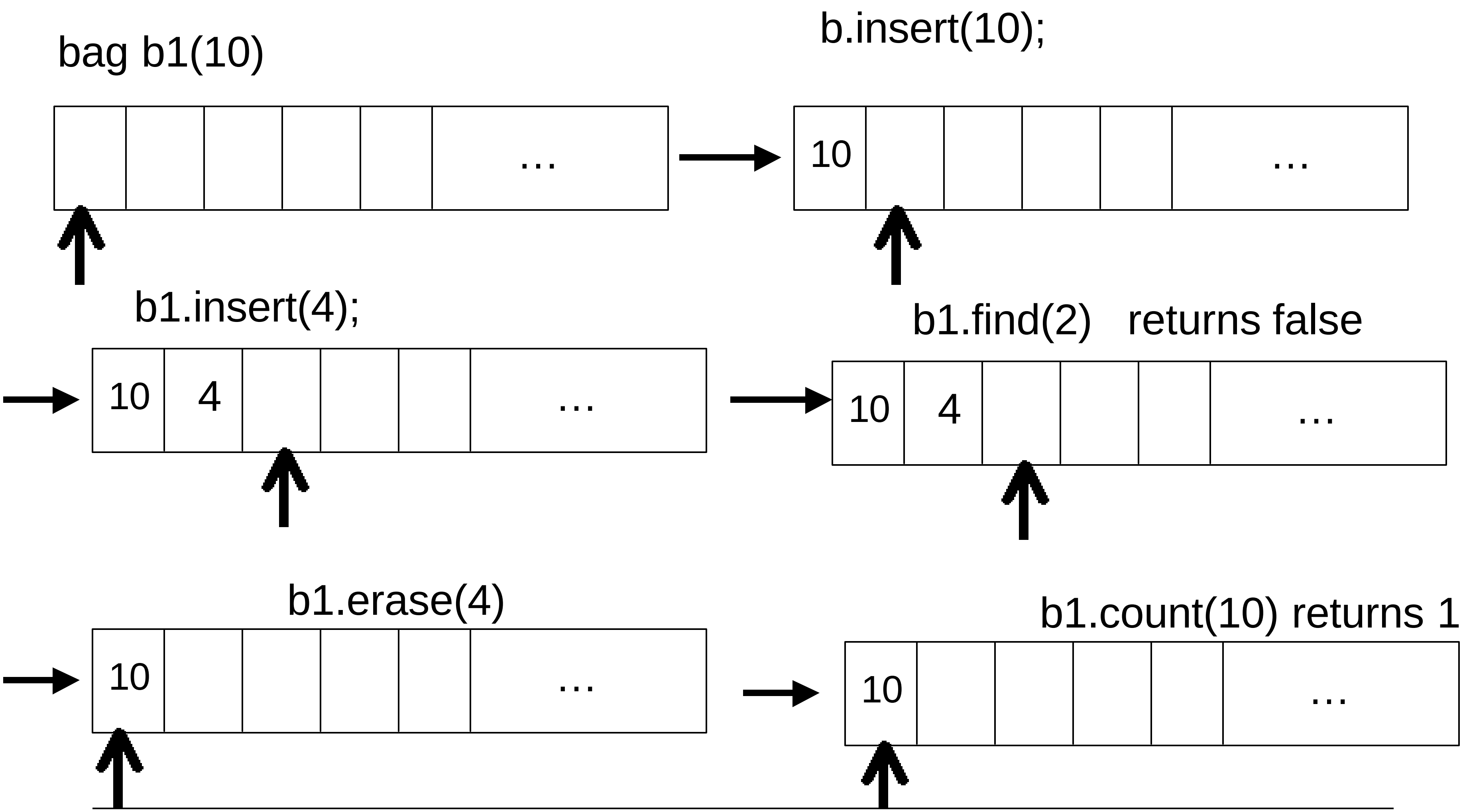
Bag Class Conceptual Example



Operation:

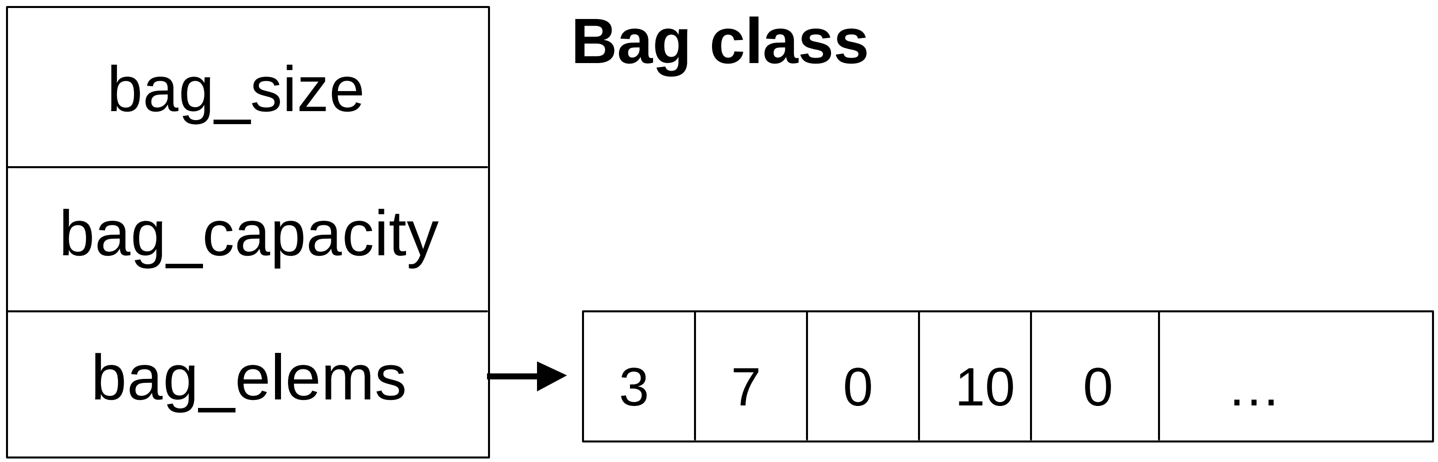
B.count(8) = 1
B.count(5) = 2
B.count(7) = 0

Using the Bag



Bag Class Design

Bag class



Wait a minute!!!
We can use vector for this!

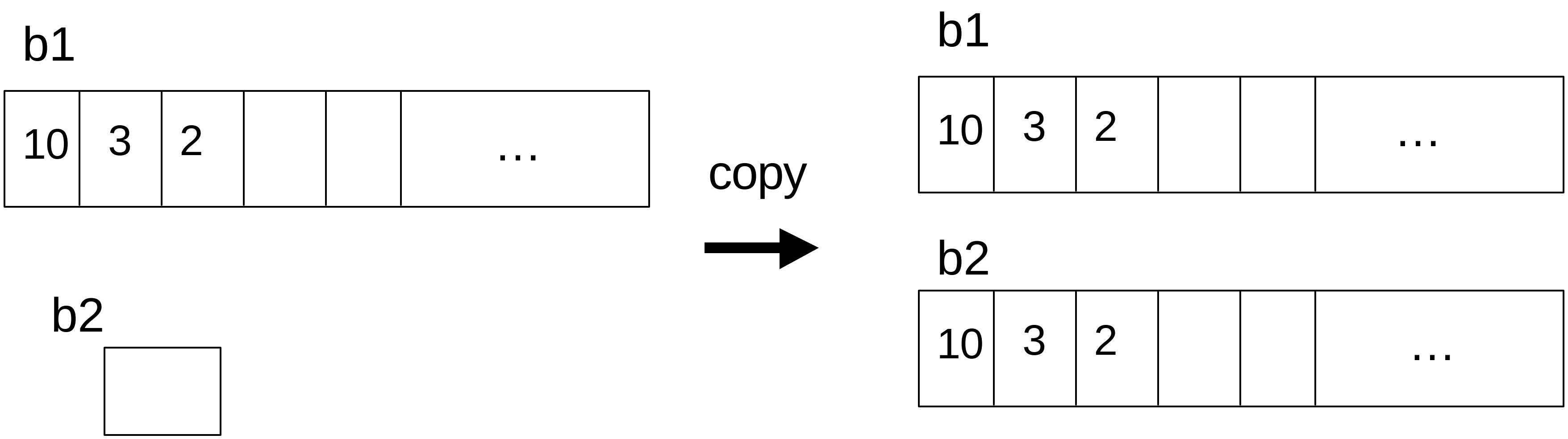
- add more semantics to implement:
  - find()
  - count()
  - insert()
  - erase()

Bag Class Methods

- Constructor
  - Make an empty bag
  - Make a bag from another bag (copy constructor)
- Accessor
  - Get current size
  - Get current capacity
  - Find an element
  - Get the count of an element
- Mutators
  - Insert a new element
  - Erase one instance of element x
  - Erase all instances of element x
- Non-member
  - Addend all elements from two bags to create a new one.

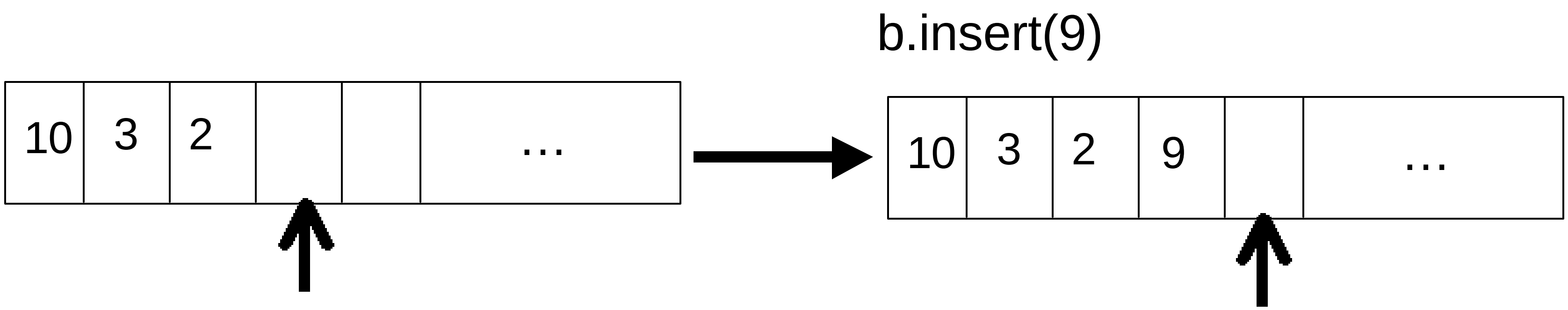
Constructors and copy methods

- Constructors initialize the vector stored in the bag
- Constructor with integer allocates a vector of size N
- Copy constructor and copy assignment simply call copy assignment operator on local vector using the vector of the argument bag.
  - Make an independent (deep) copy of the vectors



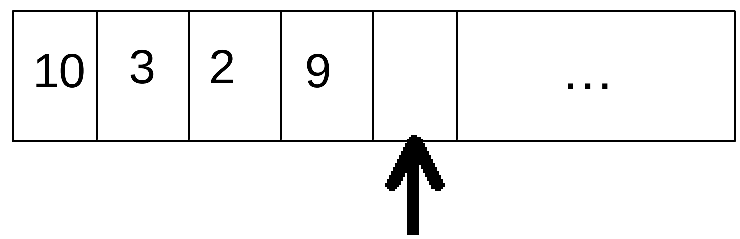
Simple ones …

- Methods that are quite trivial
  - size() – simply calls size method on vector
  - capacity() – simply class capacity method on vector
  - Operator << - prints the elements of the bag in the form:
    - { 2 3 4 5 }
- Methods that are simple but tricky
  - insert – push back a new element into the vector

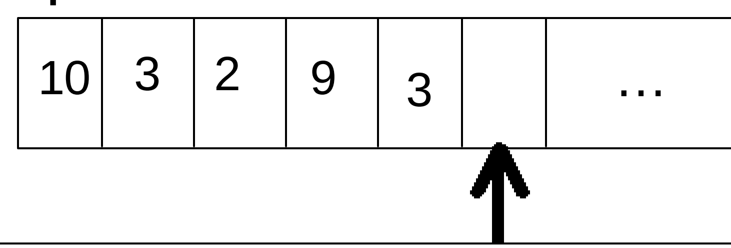


Find and count

- Find simply traverse the vector associated with the bag until it either
  - Find the element and returns true
  - Reaches the end of section in use (position size – 1) and returns false
    - Ex: b.find(9) returns true

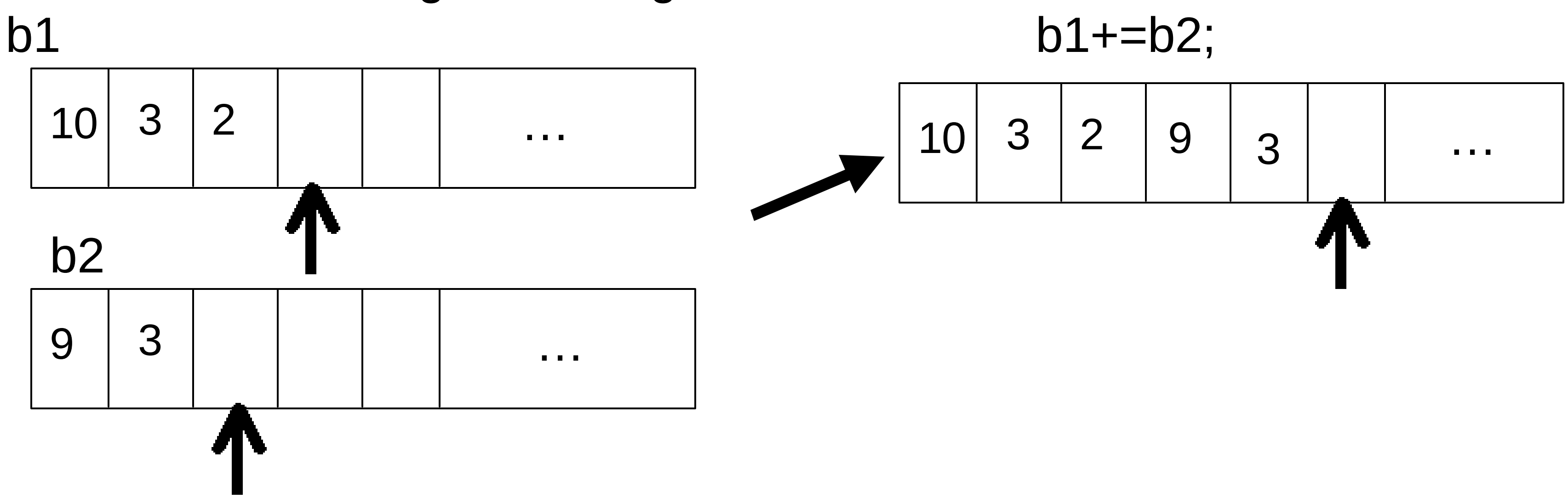


- Count traverses the array and counts the number of times it sees the element in question
    - B.count(3) returns 2

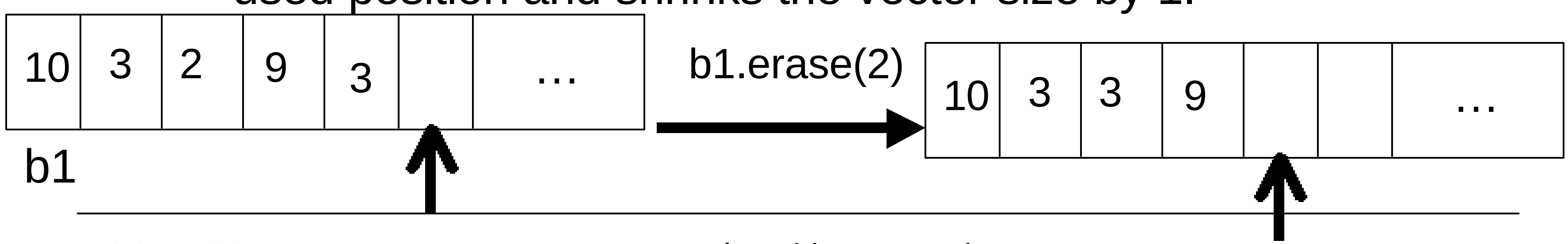


More complex operators

– Operator += - make a push back of all elements in the vector of the argument bag

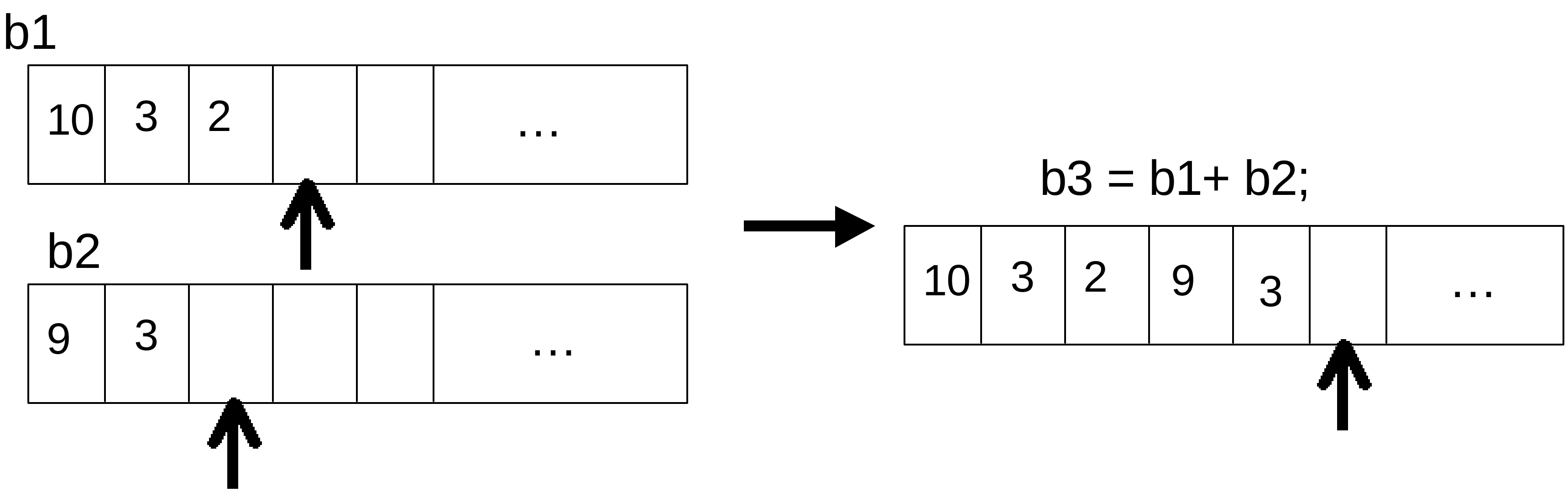


– Erase replaces the target element with the element at last used position and shrinks the vector size by 1.



Final ones

- Erase All simply calls erase until all copies of the element are delete
- Operator + is declared as non-meber. It creates a **new bag** with all elements of the two argument bags.
  - Implemented using copy constructor and += operator



Set Class Container

- Set is data structure used to store elements with the following semantics:
  - Only one copy of an element can be stored in the set S.
  - A find operation is supported to determine if an element x is present in the set S.
  - An erase operation is supported to erase an element x from the set S.
  - The following set operations are supported:
    - Union – elements that appear in at least one of two sets
    - Intersection – elements that are common to two sets
    - Difference – elements that appear in the first set but no in the second one.
    - Sub-set – determines is a set A is a subset of another set B.
- Sets are used in application in which copies are not allowed:
  - Clients in a video store, holydays in a month, social security #

On the Theory of Sets

- A set S is a collection of objects, where there are no duplicates
  - Examples
    - A = {a, b, c}
    - B = {0, 2, 4, 6, 8}
    - C = {Jose, Pedro, Ana, Luis}
- The objects that are part of a set S are called the elements of the set.
  - Notation:
    - 0 is an element of set B is written as $0 \in B$.
    - 3 is not an element of set B is written as $3 \notin B$.

Cardinality of Sets

- Sets might have
  - 0 elements – called the empty set ∅.
  - 1 elements – called a singleton
  - N elements – a set of N elements (called a finite set)
    - Ex: S = {car, plane, bike}
  - ∞ elements – an infinite number of elements (called infinite set)
    - Integers, Real,
    - Even numbers: E = {0, 2, 4, 6, 8, 10, …}
      - Dot notation means infinite number of elements
- The cardinality of a set is its number of elements
  - Notation: cardinality of S is denoted by |S|
  - Could be an integer number or infinity symbol ∞.

Cardinality of Sets (cont.)

- Some examples:
  - A = {a,b,c}
    - |A| = 3
  - R – set of real numbers
    - |R| = ∞
  - E = {0, 2, 3, 4, 6, 8, 10, …}
    - |E| = ∞
  - ∅ the empty set
    - | ∅ | = 0

Set notations and equality of Sets

- Enumeration of elements of set S
  - A = {a,b c}
  - E = {0, 2, 4, 6, 8, 10, …}
- Enumeration of the properties of the elements in S
  - E = {x : x is an even integer}
  - E = {x: x $\in$ I and x/2=0, where I is the set of integers.}
- Two sets are said to be equal if and if only they both have the same elements
  - A = {a, b, c}, B = {a, b, c}, then A = B
  - if C = {a, b, c, d}, then A $\neq$C
    - Because d $\notin$ A

Sets and Subsets

- Let A and B be two sets. B is said to be a subsets of A if and only if every member x of B is also a member of A
  - Notation: $B \subseteq A$
  - Examples:
    - A = {1, 2, 3, 4, 5, 6}, B = {1, 2}, then $B \subseteq A$
    - D = {a, e, i, o, u}, F = {a, e, i, o, u}, then $F \subseteq D$
  - If B is a subset of A, and $B \neq A$, then we call B a proper subset
    - Notation: $B \subset A$
    - A = {1, 2, 3, 4, 5, 6}, B = {1, 2}, then $B \subset A$
  - The empty set $\emptyset$ is a subset of every set, including itself
    - $\emptyset \subseteq A$, for every set A
  - If B is not a subset of A, then we write $B \not\subset A$

Set Union

- Let A and B be two sets. Then, the union of A and B, denoted by $A \cup B$ is the set of all elements x such that either $x \in A$ or $x \in B$.
  - $A \cup B = \{x: x \in A \text{ or } x \in B\}$
- Examples:
  - A = {10, 20 , 30, 40, 100}, B = {1,2 , 10, 20} then $A \cup B$ = {1, 2, 10, 20, 30, 40, 100}
  - C = {Tom, Bob, Pete}, then $C \cup \emptyset = C$
  - For every set A, $A \cup A = A$

Set Intersection

- Let A and B be two sets. Then, the intersection of A and B, denoted by $A \cap B$ is the set of all elements x such that $x \in A$ and $x \in B$.
  - $A \cap B = \{x: x \in A \text{ and } x \in B\}$
- Examples:
  - A = {10, 20 , 30, 40, 100}, B = {1,2 , 10, 20} then $A \cap B$ = {10, 20}
  - Y = {red, blue, green, black}, X = {black, white}, then $Y \cap X$ = {black}
  - E = {1, 2, 3}, M={a, b} then, $E \cap M = \emptyset$
  - C = {Tom, Bob, Pete}, then $C \cap \emptyset = \emptyset$
- For every set A, $A \cap A = A$
- Sets A and B disjoint if and only if $A \cap B = \emptyset$
  - They have nothing in common

Set Difference

- Let A and B be two sets. Then, the difference between A and B, denoted by A - B is the set of all elements x such that $x \in A$ and $x \notin B$.
  - $A - B = \{x: x \in A \text{ and } x \notin B\}$
- Examples:
  - A = {10, 20 , 30, 40, 100}, B = {1,2 , 10, 20} then A - B = {30, 40, 100}
  - Y = {red, blue, green, black}, X = {black, white}, then Y - X = {red, blue, green}
  - E = {1, 2, 3}, M={a, b} then, E - M = E
  - C = {Tom, Bob, Pete}, then C - ∅ = C
  - For every set A, A - A = ∅