

# INSIEMI E DIZIONARI

---

Angelo Di Iorio

Università di Bologna

# Insiemi

- Operazioni caratterizzanti: UNION, INTERSECTION, DIFFERENCE
- Diverse realizzazioni (e costi):

	contains lookup	insert	remove	min	Scansione	Ordine
Vettore booleano	$O(1)$	$O(1)$	$O(1)$	$O(N)$	$O(N)$	Sì
Lista non ordinata	$O(n)$	$O(1)^\dagger$	$O(n)$	$O(n)$	$O(n)$	No
Lista ordinata	$O(n)$	$O(n)$	$O(n)$	$O(1)$	$O(n)$	Sì
Hash (caso medio) memorizzazione interna	$O(1)$	$O(1)$	$O(1)$	$O(m)$	$O(m)$	No
Hash (caso medio) memorizzazione esterna	$O(1)$	$O(1)$	$O(1)$	$O(m + n)$	$O(m + n)$	No
Alberi bilanciati di ricerca	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(\log n)$	$O(n)$	Sì

$^\dagger$ Sapendo a priori che l'elemento da inserire non è già presente, altrimenti la complessità è  $O(n)$ .

# Java Collection Framework: Set e Map

Interface	Hash Table	Resizable Array	Balanced Tree	Linked List	Hash Table + Linked List
Set	HashSet		TreeSet		LinkedHashSet
List		ArrayList		LinkedList	
Deque		ArrayDeque		LinkedList	
Map	HashMap		TreeMap		LinkedHashMap

- Hash vs. Tree:
  - Implementazioni basate strutture dati diversi
  - Ordine (confronto tra elementi, ordine di inserimento)
  - Tempi di accesso diversi
  - Tempi di ristrutturazione (implementazioni basate su Hash)
  - Diversa gestione elementi null

# Dizionari (Map) in Java

- Implementare una classe Java che nel metodo `main()` trova la parola più frequente in un testo letto da file
- Nota: il seguente frammento di codice usa la classe Java `Scanner` (built-in) per leggere il contenuto di un file (senza controlli sull'esistenza!) e dividerlo in token, eliminando la punteggiatura, per poi analizzarli usando un iteratore:

```
Scanner scanner = new Scanner(  
    new File("src/tests/maps/", "notes.txt"));  
Scanner doc = scanner.useDelimiter("[^a-zA-Z]+");  
while (doc.hasNext( )) { ... doc.next() ... }
```

# Insiemi (Set) in Java

- Aggiungere il codice per costruire l'insieme di parole di lunghezza minore di 10 e quello di parole che iniziano per "a"
- Usare i metodi dell'interfaccia `Set` per estrarre da questi insiemi le parole che iniziano per "a" ed hanno una lunghezza minore di 10
  - Oppure le parole che iniziano per "a" oppure hanno una lunghezza minore di 10