Generic .NET Collections - Comparison Chart

Sequential Processing

Queue<T>: FIFO queue. Use for tasks in order.

Stack<T>: LIFO stack. Use for undo/backtrack.

PriorityQueue<TElement, TPriority>: Items with priority. Lower value = higher priority.

Set-like Collections (Uniqueness)

HashSet<T>: Unordered unique values. Fast lookup.

SortedSet<T>: Sorted unique values.

Key - Value Mapping

SortedDictionary<TKey, TValue>: Dictionary with sorted keys. Efficient inserts.

SortedList<TKey, TValue>: Sorted key-value pairs with index-based access.

KeyValuePair<TKey, TValue>: Lightweight (key, value) pair structure.

Thread-safe Collections

SynchronizedCollection<T>: Thread-safe list. Use when multiple threads access items.

Quick Recommendations

- Need a simple queue? Use Queue<T>
- Priority jobs? Use PriorityQueue<TElement, TPriority>
- Unique unordered items? Use HashSet<T>
- Unique sorted items? Use SortedSet<T>
- Key-value lookup with sorting? Use SortedDictionary or SortedList
- Multiple threads? Use SynchronizedCollection<T>