Worksheet: Abstract Collection

The `Collection` interface is defined as follows (the default methods declared in this interface are not shown):

```
interface Collection<E> extends Iterable<E> {
      Iterator<E> iterator();
      boolean add(E x);
      int size();
      boolean isEmpty();
      boolean contains(Object x);
      boolean containsAll(Collection<?> c);
      boolean addAll(Collection<? extends E> c);
      boolean remove(Object x);
      boolean removeAll(Collection<?> c);
      boolean retainAll(Collection<?> c);
      void clear();
      Object[] toArray();
      <T> T[] toArray(T[] a);
}
```

Tasks:

1. Implement an abstract class `AbstractCollection` in which as many as possible methods declared in the `Collection` interface are provided as concrete methods (or alternatively implement as many as possible methods declared in the `Collection` interface as Java 8 default methods).

   ```
   public abstract class AbstractCollection<E> implements Collection<E> {
         …
   }
   ```

2. Which methods *cannot* be implemented in the abstract class (or as Java 8 default methods) in terms of the other methods? In other words: For which methods is information about the data structure used to store the elements necessary?
   Hint: Only two methods cannot be implemented!

3. Which of the methods which you could implement should be overridden by a concrete implementation? Justify your answer!

Note that the abstract class has *no information whatsoever* about the data structure used to store the elements (i.e. whether an array, a list or a tree, etc. is used).