Binary Search Worksheet

The following table of numbers represents the Fibonacci sequence

1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

The general algorithm:

```
private boolean binarySearch(int first, int last, T desiredItem)
   {
      boolean found;
      int mid = (first + last) / 2;

      if (first > last)
         found = false;
      else if (desiredItem.equals(list[mid]))
         found = true;
      else if (desiredItem.compareTo(list[mid]) < 0)
         found = binarySearch(first, mid - 1, desiredItem);
      else
         found = binarySearch(mid + 1, last, desiredItem);

      return found;
   }
```

The initial call

```
     binarySearch(0, numberOfEntries - 1, anEntry);
```

As a class, we will trace through the algorithm for

```
     binarySearch(0, numberOfEntries - 1, 5);
```

On your own, trace through the algorithm for

```
(1) binarySearch(0, numberOfEntries - 1, 34);
```

(2) A number that doesn’t exist in the list.