### 3.4.3 Comparison

Table 3.22 summarizes the efficiency of the three array searching algorithms.

For an unsorted array, we have no alternative but to use linear search.

For a sorted array, binary search is clearly superior to linear search. Our analyses have shown that binary search has time efficiency $O(\log n)$ whereas linear search has time efficiency $O(n)$. Binary search should be preferred unless $n$ is known to be small.

The intuition behind the binary search algorithm's superiority is clear. Compare the subarrays still to be searched (shown unshaded) in Figures 3.14 and Figure 3.20. Each iteration of the linear search algorithm removes only one component from the unsearched subarray, whereas each iteration of the binary search algorithm removes about half of the unsearched subarray's components.

An everyday situation illustrates the point vividly. When we search a 1000-page dictionary or telephone directory using binary search, we will find the target entry in 10 iterations *at most* (the number of pages still to be searched being successively 1000, 500, 250, 125, 62, 31, 15, 7, 3, and 1). If we were foolish enough to use linear search, we

```
static int binarySearch (Comparable target,
                         Comparable[] a, int left, int right) {
// Find which (if any) component of a[left...right] equals target
// (where a is sorted).
   int l = left, r = right;
   while (l <= r) {
      int m = (l + r)/2;
      int comp = target.compareTo(a[m]);
      if (comp == 0)
         return m;
      else if (comp < 0)
         r = m - 1;
      else  // comp > 0
         l = m + 1;
   }
   return NONE;
}
```

**Program 3.21** Implementation of the array binary search algorithm as a Java method.

**Table 3.22** Efficiency of array searching algorithms.

| Algorithm | No. of comparisons (approx.) | Time complexity |
|---|---|---|
| Unsorted array linear search | $n/2$ (successful)<br>$n$ (unsuccessful) | $O(n)$ |
| Sorted array linear search | $n/2$ | $O(n)$ |
| Sorted array binary search | $\log_2 n$ | $O(\log n)$ |