Figure 23.2 Another way of searching—*binary* search.

We can do much better when searching *sorted* lists.

i	**list[i]**
0	Adams
1	Biggles
2	Cassidy
3	Clarke
4	Evans
5	Jackson
6	Morrison
7	Pancer

This is a *binary search*. Given a sorted list, the basic idea is to do the following until only one item is left:

—divide the list in half
—look at the item in the middle (i.e. *COMPARISON*)
—discard the half of the list that cannot contain the item

The # of comparisons is the # of times the list can be divided in half before only one item is left.

For $N = 1,000,000$, this is a little under 20!
— this is a *huge* improvement over linear search

Binary search is acceptable, linear search is not.

- linear search is $\mathcal{O}(N)$
- binary search is $\mathcal{O}(\log N)$

And what about sorting?

- better algorithms exist, but are beyond the scope of this course
- for example, *merge* sort is $\mathcal{O}(N \log N)$ instead of $\mathcal{O}(N^2)$
- still worse than searching, but makes sorting doable on large lists