# How do we solve problems recursively?

Imagine the worst case possible: the thing you're looking for isn't in the list (100,000 elements).

1. Sequential search:
   You will look at all 100,000 elements before being able to determine the value isn't there.

2. Binary search (example run through):
   1. Look at index `50,000`, too big, search in `[0, 49,999]`
   2. Look at index `25,000`, too big, search in `[0, 24,999]`
   3. Look at index `12,500`, too small, search in `[12,501, 24,999]`
   4. Look at index `18,750`, too small, search in `[18,751, 24,999]`
   5. Look at index `21,875`, too big, search in `[18,751, 21,875]`
   6. Look at index `20,313`, too big, search in `[18,751, 20,312]`
   7. Look at index `19,531`, too small, search in `[19,532, 20,312]`
   8. Look at index `19,922`, too big, search in `[19,532, 19,921]`
   9. Look at index `19,726`, too big, search in `[19,532, 19,725]`
   10. Look at index `19,629`, too big, search in `[19,532, 19,628]`
   11. Look at index `19,580`, too big, search in `[19,532, 19,579]`
   12. Look at index `19,555`, too small, search in `[19,556, 19,579]`
   13. Look at index `19,567`, too small, search in `[19,568, 19,579]`
   14. Look at index `19,573`, too small, search in `[19,574, 19,579]`
   15. Look at index `19,576`, too small, search in `[19,577, 19,579]`
   16. Look at index `19,578`, too small, search in `[19,579, 19,579]`
   17. Look at index `19,579`, too small, search in `[19,580, 19,579]` —- WAIT A SECOND!

It took 17 passes to figure out that the ***key*** was not in the list. That is substantially faster than 100,000 passes!

# How do we solve problems recursively?