

Linear vs Binary

Linear Search vs Binary Search (aka analog vs digital)

A [linear search](#) looks down a list, one item at a time, without jumping. In complexity terms this is an $O(n)$ search - the time taken to search the list gets bigger at the same rate as the list does.

A [binary search](#) is when you start with the middle of a sorted list, and see whether that's greater than or less than the value you're looking for, which determines whether the value is in the first or second half of the list. Jump to the half way through the sublist, and compare again etc.

This is pretty much how humans typically look up a word in a dictionary (although we use better heuristics, obviously - if you're looking for "cat" you don't start off at "M")

In complexity terms this is an $O(\log n)$ search - the number of search operations grows more slowly than the list does, because you're halving the "search space" with each operation.

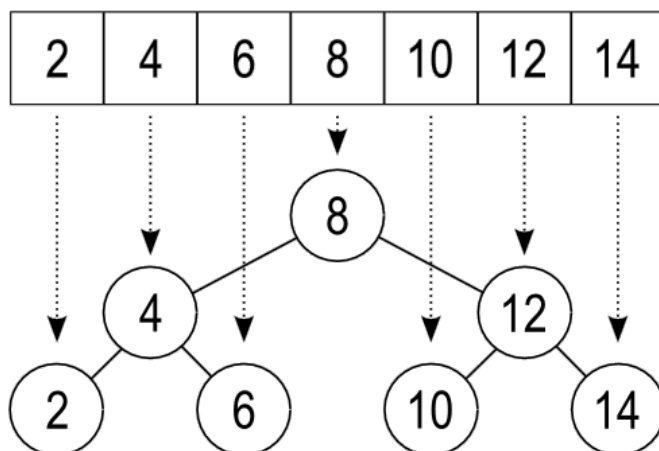
With a partner, determine whether a linear search or a binary search is more efficient.

Step 1: Your partner chooses a concept in a textbook but does not tell you what it is or where it is.

Step 2: You use a linear search to find the concept. Count how many questions it took to find the answer.

Step 3. Repeat step 1 and find a different concept.

Step 4. Find the concept but use a binary search. (Count how many questions it took to find the answer.



Questions to Reflect:

1. In which cases would a linear search algorithm be more efficient?
2. In which cases would a binary search be more efficient?
3. On average, which would be the most efficient if used 100% of the time?