

Big Data, Reading

"Big data" is a big issue in data structures. You should have already learned techniques for searching and sorting arrays using for-loops, like this:

searching an array for a match

```
int a[CAP];
...

int index;
for (index = 0; index < CAP; index++)
    if (a[index] == matchThisValue)
        break;

if (index < CAP)
    cout << "Found at index " << index << endl;
else
    cout << "Not found\n";
```

sorting an array lo-to-hi

```
int a[CAP];
...

for (int i = 0; i < CAP; i++)
    for (int j = i + 1; j < CAP; j++)
        if (a[j] < a[i])
            swap(a[i], a[j]);
```

You probably never noticed the inefficiencies of these methods, because on the computers you've been using, and with the capacities of your arrays, these loops run so fast that you don't even notice a delay. But all that's about to change when we get to "big data", and a challenge for students of data structures is how to overcome what's coming.

Big data is about "data sets so large or complex that traditional... data processing applications are inadequate" (ref [wikipedia.org](https://en.wikipedia.org/wiki/Big_data) (https://en.wikipedia.org/wiki/Big_data)). "Traditional data processing" techniques are what you've been using when you apply the loops presented above. "Inadequate" refers to the running time of your computer applying these techniques when the data set is large.

Try to imagine what it's like for Google to search its databases of billions of entries when you request a search, and think about how fast a result gets returned to you. How *do* they *do that*? To be sure, it's not the loop you see above, with a CAP in the billions. And when they sort the result by relevance as your list of results is presented to you, to be sure, there are *not* using the nested for-loop that you see above. Because if they did, your queries would take days to answer, and Google would never have become the successful company it is today.

Google must know something we don't!