CS 112 – Introduction to Computing II

Motivation: Two algorithms for Searching an Array

* How do we think scientifically about the programs we write? Mostly this is by analyzing how they use resources(time, space, hardware, power, other algorithms). We will focus in this class on understanding the running time of algorithms.
* Example:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 78 | 25 | 2 | 15 | 26 | 38 | 7 | 45 | 12 | 19 |

* + How would we determine if a given integer, say 19, is in the list
  + This problem is called “sequential search” or “linear search.”
  + We go over the list one by one and see if we can find the integer in the list.
* How would we analyze this algorithm?
* We are basically interested in how long it takes to find an arbitrary member of the list…
* Here are the kinds of questions we want to answer:
  + How many “basic operations” (e.g. comparing one integer to another) does it take to find the integer (or not), expressed as a function of N = number of data items.
    - In the worst case?
    - In the best case?
    - In the average case?
* Now let’s consider how things change when we sort the list into ascending order:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2 | 7 | 12 | 15 | 19 | 25 | 26 | 38 | 45 | 78 |

* Now how would we determine if a given integer, say 15, is in the list
* We can use “binary search”
* Find the middle, if it is equal to the number; if not, to see if it is smaller or larger. Do the last step