

Sequential Search versus Binary Search

- 1) Generate an ordered list of 1000 integers (1,2,3,..., 1000)
- 2) Generate a random key $0 \leq \text{key} \leq 1000$
- 3) Use a sequential search to find the key in the list using a counter to keep track of how many times it took to locate it.
- 4) Use a binary search to find the key in the list and record the number of tries it took to locate it (n).
- 5) Print the percentage gain/loss for binary over sequential search
- 6) Repeat 6 times.

Example of percentage gain/loss:

In January Dylan worked a total of 35 hours, in February he worked 45.5 hours – by what percentage did Dylan’s working hours increase in February?

To tackle this problem first we calculate the difference in hours between the new and old numbers. $45.5 - 35 \text{ hours} = 10.5 \text{ hours}$. We can see that Dylan worked 10.5 hours more in February than he did in January – this is his **increase**. To work out the increase as a percentage it is now necessary to divide the increase by the original (January) number:

$10.5 \div 35 = 0.3$ (See our [division](#) page for instruction and examples of division.)

Finally, to get the percentage we multiply the answer by 100. This simply means moving the decimal place two columns to the right.

$0.3 \times 100 = 30$

Dylan therefore worked 30% more hours in February than he did in January.