C++ code for a linear search:

```
template <typename Arr, typename Elem>
long linsearch(const Arr & a, long MAX, const Elem & f)
{
    long i = 0;
    while (i != MAX && a[i] != f)
    {
        i++;
    }
    return (i == MAX ? –1 : i);
}
```

Generalizing the algorithm:

```
LinearSearch(array, n, find)
{
    i = 1
    while (i <= n AND array[i] != find)
    {
        i = i+1
    }
    if (i > n)
    {
        i = 0
    }
    return i
}
```

We take the comparison as our main operation (since the goal is to locate something, you must compare to realize you've found it). We count only one of the two comparisons in the loop condition since it would simply multiply things by 2. (We don't really worry about the short-circuit property of the **AND** since that only applies in specific computer languages – not in generic algorithms.) Counting we find:

Best	Worst	Average	Found at (i)	Count
1	0	1/(n+1)	1	2
0	0	1/(n+1)	2	3
…	…	…	…	…
0	0	1/(n+1)	n	n+1
0	1	1/(n+1)	None (n+1)	n+2

Best case time is 2 which is O(1) (or constant time). Worst case is n+2 which is O(n) (or linear time).

We see that the general formula for c(i) is i+1. For the average case, p(i) is always 1/(n+1).

To find the average time, we perform our standard summation:

$$A(n) = \sum_{i=1}^{n+1} p(i)c(i) = \frac{1}{n+1}\left(\sum_{i=1}^{n} (i+1) + (n+2) \right)$$

So, for all the internal locations (where **`find`** is found) we have:

$$\frac{1}{n+1}\sum_{i=1}^{n} i+1 = \frac{1}{n+1}\left(\sum_{i=1}^{n} i + \sum_{i=1}^{n} 1 \right)$$

$$= \frac{1}{n+1}\left(\frac{n(n+1)}{2} + n \right)$$

$$= \frac{\left(n(n+1)+2n\right)}{2(n+1)}$$

$$= \frac{n}{2} + \frac{n}{n+1}$$

And for the last position (where **`find`** is not found), we have: $\frac{n+2}{n+1}$

Together they make:

$$\frac{n}{2} + \frac{2n+2}{n+1} = \frac{n}{2} + \frac{2(n+1)}{n+1} = \frac{n}{2} + 2$$

So, dropping the constant, we have A(n) = n/2 or "the average time for linear search is n/2".

Now to apply our Big-O litmus tests. Algebraically we choose:

$$C = 3 \text{ and } n_0 = 0$$

So that for n strictly greater than n_0 we have:

$$\left|\frac{n}{2} + 2\right| \leq 3|n|$$

Since all of our n values, the absolute value is the same as the value we have:

$$\frac{n}{2} + 2 \leq 3n$$

Checking:

$$0 \leq 3n - \frac{n}{2} - 2$$

$$\leq \frac{5n}{2} - 2$$

$$\leq 5n - 4$$

$$4 \leq 5n$$

$$\frac{4}{5} \leq n$$

Which is true since we chose n > 0 and all our n values are integral. Therefore:
$A(n) \in O(n)$

By limits (with a little L'Hopital) we get:

$$\lim_{n \to \infty} \frac{n/2+2}{n} = \lim_{n \to \infty} \frac{1/2}{1} = \frac{1}{2}$$

And since ½ is strictly less than ∞, we again have that: $A(n) \in O(n)$