
Basic Operation

The amount of time it takes to execute a basic operation is not related to the input size. That means we can *count* the number of basic operations and use that count as a proxy for the actual running time.

Below are common basic operations.

1. Assignment statements
   Examples

   ```
   x = 5
   x = -8526
   ```

2. Method calls

3. Arithmetic operations
   Examples

   ```
   x + y
   (x * x) + (y * y)
   ```

4. Comparisons
   Examples

   ```
   x <= 8
   x == max
   ```

5. Array/List indexing
   Example

   ```
   L[3]
   array[12]
   ```

6. Following an object reference

7. Returning from a method or function

Count the number of times the basic operations are executed; this should be proportional to the input size. As the input size gets bigger, the number of times the basic operation executed grows. For example, if the input size doubles, the number of basic operation executed may also double. It might also be the case that the number of basic operations executed stays the same or more than doubles.

Input Size

For a given instance of a problem, we designate the input size as n. For algorithms operating on lists or arrays, the input size is commonly the number of elements in the list or array.

If the input is a number, the input size is the number of bits needed to represent the input.

Time Complexity, T(n)

"Every case" Time complexity

For input size of n, the running time is the same for no matter what is the input.

$$T(n) = n$$

Example: Sum the elements of list, L

```
sum = 0
for each List item x
        sum = sum + x
return sum
```

Exchange Sort

```
Input: L, a list of numbers
Output: L in sorted order

n = number of elements in L (this is the input size)
for i from 0 to n-1
    for j from i+1 to n-1
       if L[j] < L[i]
          Swap L[i] and L[j]
```

The basic operation for exchange sort is the comparison `L[j] < L[i]`

Trace of Exchange Sort on list `[3, 2, 5, 4]`

```
                              [3, 2, 5, 4]
i = 0
   j = 1 swap L[i] & L[j]     [2, 3, 5, 4]
   j = 2 No swap
   j = 3 No swap

i = 1
   j = 2 No Swap
   j = 3 No Swap

i = 2
   j = 3 Swap L[i] & L[j]     [2, 3, 4, 5]

i = 3
   Inner loop doesn't execute

Outer loop terminates
```

How many times does the basic operation execute for input size n?

The outer loop executes n times, with i going from 0 to $n-1$.

For each execution of the outer loop, the inner loop has j going from $i+1$ to $n-1$.

The basic operation is in the inner loop. So we can count how many times the inner loop executes, using as an example $n = 5$. Then i goes from 0 to 4, and for each value of i, j goes from $i+1$ to 4.

```
i = 0
    j =  1    2    3    4

i = 1
    j =       2    3    4

i = 2
    j =            3    4

i = 3
    j =                 4

i = 4
    j N/A inner loop doesn't execute
```

So for $n = 5$, the basic operation executes 10 times.
In general, for an input size of n, the number times the basic operation executes is

$$(n-1) + (n-2) + \ldots + (n-(n-1))$$

Since $n - (n-1) = 1$, we can rewrite this summation as

$$1 + 2 + \ldots + (n-1)$$

Using summation notation

$$\sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}$$

Patterns of nested loop

Option 1:

```
for i from START to END
    for j from START to END

    x x x x x x x x
    x x x x x x x x
    x x x x x x x x
    x x x x x x x x
    x x x x x x x x
    x x x x x x x x
    x x x x x x x x
    x x x x x x x x
```

Running time is n^2

Option 2:

```
for i from START to END
    for j from START to i

    x
    x x
    x x x
    x x x x
    x x x x x
    x x x x x x
    x x x x x x x
    x x x x x x x x
```

Running time is $(n^2)/2$

Option 3:

```
for i from START to END
    for j from i to END

    x x x x x x x
      x x x x x x
        x x x x x
          x x x x
            x x x
              x x
                x
```

Running time is $(n^2)/2$

Option 4:

```
for i from START to END
    for j START to END - i

    x x x x x x x
    x x x x x x
    x x x x x
    x x x x
    x x x
    x x
    x
```

Running time is $(n^2)/2$

Option 5:

```
for i from START to END
    for j END-i to END - 1

                x
              x x
            x x x
          x x x x
        x x x x x
      x x x x x x
    x x x x x x x
```

Running time is $(n^2)/2$