complexity_analysis

August 26, 2021

1 The Complexity of Algorithims

Motivation Sometimes it is necessary to have a precise understanding of the complexity of an algorithm. In order to obtain this understanding we could proceed as follows:

We implement the algorithm in a given programming language.

We count how many additions, multiplications, assignments, etc.~are needed

for an input of a given size. Additionally, we have to count all storage accesses.

We look up the amount of time that is needed for the different operations in the processor handbook.

Using the information discovered in the previous two steps we predict the running time of our algorithm for given input.

1.0.1 Naive Approach

$S = 1 + 2 + 3 + ... + n$

[8]:
```
import time


def sum_of_n_2(n):
    start = time.time()

    the_sum = 0
    for i in range(1, n + 1):
        the_sum = the_sum + i

    end = time.time()

    return the_sum, end - start

for i in range(5):
    print("Sum is %d required %10.6f seconds" % sum_of_n_2(10000))
```

```
Sum is 50005000 required   0.000823 seconds
Sum is 50005000 required   0.000839 seconds
```

```
Sum is 50005000 required   0.000982 seconds
Sum is 50005000 required   0.000876 seconds
Sum is 50005000 required   0.000861 seconds
```

1.0.2 Alternative Approach

$S = 1 + 2 + 3 + ... + n$

$S = n + (n-1) + (n-2) + ... + 1$

$2S = (n+1) + (n+1) + ... + (n+1) = n * (n+1)$

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

[11]:
```
def sum_of_n_3(n):
    start = time.time()

    the_sum = ((n+1)*n)/2.0

    end = time.time()

    return the_sum, end - start

for i in range(5):
    print("V2: Sum is %d required %10.6f seconds" % sum_of_n_3(10000))
```

```
V2: Sum is 50005000 required   0.000002 seconds
V2: Sum is 50005000 required   0.000001 seconds
V2: Sum is 50005000 required   0.000001 seconds
V2: Sum is 50005000 required   0.000001 seconds
V2: Sum is 50005000 required   0.000000 seconds
```

[ ]:

[ ]: