

# Counting DNA Nucleotides

Jan Emmanuel Samson

2024-07-28

## Problem

A string is simply an ordered collection of symbols selected from some **alphabet** and formed into a word; the **length** of a string is the number of symbols that it contains.

An example if a length 21 DNA string (whose alphabet contains the symbols 'A', 'C', 'G', and 'T') is "AT-GCTTCAGAAAGGTCTTACG."

- **Given:** A DNA string *s* of length at most 100 nt.
- **Return:** Four integers (separated by spaces) counting the respective number of times that the symbol 'A', 'C', 'G', and 'T' occurs in *s*.

## Sample Dataset

```
AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAGAGTGTCTGATAGCAGC
```

## Sample Output

```
20 12 17 21
```

## Intuition

Represent the DNA sequence as a string and initialize an empty dictionary with unique bases as keys with a value of zero. Iterate over the sequence one base at a time. If the current base is not a dictionary key, then it is a non-canonical base which should be ignored. Otherwise, index the dictionary by the current base and increment its value by 1.

## Solution

```
def count_nucleotides(seq: str, alphabet: str='ACGT') -> dict[str, int]:  
    """Return the counts of A, C, G, and T in a DNA sequence."""  
    counts = {base: 0 for base in alphabet}  
    for base in seq.upper():  
        if base in counts:  
            counts[base] += 1  
    return counts.values()
```

```
seq = "AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAGAGTGTCTGATAGCAGC"  
result = count_nucleotides(seq)  
print(*result)
```

```
20 12 17 21
```