Substring, Subarray, and Subsequence Explained

1. Substring
A substring is a continuous part of a string. Characters must appear in the same order and without skipping any character.
Example: "abcde"
Substrings: "a", "ab", "abc", "abcd", "abcde", "b", "bc", "bcd", "bcde", ...
■ "ace" is not a substring because characters are not continuous.
Number of substrings for a string of length n = n * (n + 1) / 2
Time Complexity (to generate all substrings): O(n²)

2. Subarray
A subarray is similar to a substring but applies to arrays (continuous elements of the array).
Example: [1, 2, 3]
Subarrays: [1], [2], [3], [1, 2], [2, 3], [1, 2, 3]
Number of subarrays for array of length n = n * (n + 1) / 2
Time Complexity: O(n²)

3. Subsequence
A subsequence is formed by deleting zero or more elements without changing the relative order.
Example: "abc"
Subsequences: "a", "b", "c", "ab", "ac", "bc", "abc"
Number of subsequences for string of length n = 2■ - 1
Time Complexity (to generate all subsequences): O(2■)

C++ Implementation

```
// C++ Code
#include <iostream>
#include <vector>
#include <string>
using namespace std;

void substrings(string s) {
    int n = s.length();
    cout << "Substrings:\n";
    for (int i = 0; i < n; i++)
        for (int j = i; j < n; j++)
            cout << s.substr(i, j - i + 1) << " ";
    cout << "\n";
}

void subsequences(string s, string out="", int index=0) {
    if (index == s.size()) {
        if (!out.empty()) cout << out << " ";
        return;
    }
    subsequences(s, out, index + 1);
    subsequences(s, out + s[index], index + 1);
}

int main() {
    string s = "abc";
    substrings(s);
    cout << "Subsequences:\n";
    subsequences(s);
}
```

Java Implementation

```
// Java Code
```

```
public class Main {
    static void substrings(String s) {
        System.out.println("Substrings:");
        for (int i = 0; i < s.length(); i++)
            for (int j = i + 1; j <= s.length(); j++)
                System.out.print(s.substring(i, j) + " ");
        System.out.println();
    }

    static void subsequences(String s, String out, int index) {
        if (index == s.length()) {
            if (!out.isEmpty()) System.out.print(out + " ");
            return;
        }
        subsequences(s, out, index + 1);
        subsequences(s, out + s.charAt(index), index + 1);
    }

    public static void main(String[] args) {
        String s = "abc";
        substrings(s);
        System.out.println("Subsequences:");
        subsequences(s, "", 0);
    }
}
```

Python Implementation

```
# Python Code
def substrings(s):
    print("Substrings:")
    for i in range(len(s)):
        for j in range(i + 1, len(s) + 1):
            print(s[i:j], end=" ")
    print()

def subsequences(s, out="", index=0):
    if index == len(s):
        if out:
            print(out, end=" ")
        return
    subsequences(s, out, index + 1)
    subsequences(s, out + s[index], index + 1)

s = "abc"
substrings(s)
print("Subsequences:")
subsequences(s)
```