Dynamic Programming — Palindromes

Competitive Algorithmic Programming

"If you want to go fast, go alone; if you want to go far, go together." – possibly an African proverb.

Part 1 - Example

- Consider the word `ababbab`. Find the palindromes!

	a	b	a	b	b	a	b
a							
b							
a							
b							
b							
a							
b							

Part 2 - Code

Kotlin Code

```kotlin
fun main() {
    val name = "ababbab"
    var count = 0
    val dp = Array(name.length) { Array(name.length) { false } }

    for (i in 0..name.length-1)
        dp[i][i] = true
    for (i in 1..name.length-1)
        if (name[i-1] == name[i]) {
            dp[i-1][i] = true
            count++

    for (gap in 2..name.length-2) {
        var i = 0
        for (j in gap..name.length-1) {
            i++
            if (name[i] == name[j] && dp[i+1][j-1]) {
                count++
                dp[i][j] = true
            }
        }
    }
    println("count = $count")
}
```

C++ Code

```cpp
int numPalindromes(stirng s) {
  int i,j,gap,count;
  vvb dp(s.length(),vb(s.length()),false);

  count = 0;
  for(i=0; i<s.length(); ++i)
      dp[i][i] = true; // one character palindroms

  // base casee: two character palindromes
  for(i=1; i<s.length(); ++i)
      if (s[i-1] == s[i]) {
         dp[i-1][i] = true;
         ++count;
    }

  for(gap=2; gap<s.length()-1; ++gap)
     for(j=gap, i=0; j<s.length(); ++i, ++j)
        if (s[i] == s[j] && dp[i+1][j-1]) {
            ++count;
            dp[i][j] = true;
         }
  return count;
}
```