

```

def generate_string(alphabet, length, forbidden, prefix=""):
    if length == 0:
        return [""]
    else:
        result = []
        for char in alphabet:
            new_prefix = prefix + char
            if not any(str in new_prefix for str in forbidden):
                for string in generate_string(
                    alphabet, length - 1, forbidden, new_prefix
                ):
                    result.append(char + string)
        return result

```

مقدار *prefix* در این برنامه رشته‌ای است که با پیمایش از ریشه تا راس کنونی ساخته شده است و *result* از اضافه کردن مقدار گره کنونی به هر یک رشته‌های ساخته شده توسط زیردرخت‌های این گره به دست می‌آید، بقیه برنامه همان چیزی است که در الگوریتم قسمت قبل توضیح داده شده است.

نتیجه آزمایش برنامه برای ورودی‌های مختلف به شرح زیر است:

```

alphabet = 01, length = 5
forbidden = ['001', '11']
Generated 6 strings
strings = ['00000', '01000', '01010', '10000', '10100', '10101']
3.5e-05 seconds

```

```

alphabet = 01, length = 20
forbidden = ['0101', '1100']
Generated 67483 strings
0.18099 seconds

```

```

alphabet = abc, length = 5
forbidden = ['aa', 'acb', 'abcabc', 'b']
Generated 13 strings
strings = ['acaca', 'acacc', 'accac', 'accca', 'acccc', 'cacac', 'cacca', 'cacc',
0.000417 seconds

```

```

alphabet = abc, length = 25
forbidden = ['aabb', 'b', 'ccaa']
Generated 5600910 strings
16.842465 seconds

```

```

alphabet = 123456, length = 12
forbidden = ['1', '2', '3', '45']
Generated 121393 strings
0.333522 seconds

```

```

alphabet = abcdefghijklmnopqrstuvwxyz, length = 5
forbidden = ['hello', 'ae', 'ua', 'akw']

```

Generated 11741131 strings
8.276327 seconds