

پروژه دوم

طراحی الگوریتم دکتر احمدی

باند، جیمز باند!

علیرضا فرزانه

راه ساده این است که رشته ورودی را به با استفاده از یک اندیس به دو قسمت (قسمت راست آن اندیس و قسمت چپ آن اندیس) تقسیم کنیم، قسمت چپ را در دیکشنری داده شده چک کنیم و قسمت راست را بازگشتی تکرار کنیم.

اما در این راه ساده، تعداد بسیاری زیر مسئله مشابه مکرراً حل می شود که با استفاده از dynamic programming قابل بهبود است.

با استفاده از dynamic programming، رویکرد پایین به بالا و ذخیره سازی زیر مسئله ها می توانیم مسئله ای از پیچیدگی نمایی را به مسئله ای از $O(n^2)$ تبدیل کنیم. همچنین پیچیدگی حافظه نیز از $O(n)$ است که n طول رشته ورودی است.

با دو حلقه تو در تو و ذخیره سازی حالات به دست آمده در یک آرایه مسئله حل می شود.

```
for 0 < i < s.length + 1; i++
```

```
for 0 <= j < i; j++
```

```
if T[j] && is_in_dictionary(s.substring(j, i))
```

```
T[i] = True
```

```
Break
```

```
return T[s.length]
```

به اینگونه در آرایه T اگر زیر رشته از ۰ تا i قابل تقسیم باشد، حالت تقسیم شدن در T[i] ذخیره می شود.

در نهایت اگر کل رشته قابل تقسیم باشد خانه آخر آرایه نشان دهنده حالت تقسیم شدن کل رشته است.

نمونه ورودی ۱:

BROWNFOXJUMPEDOVERTHEBLACKDOG

نمونه خروجی ۱:

brown fox jumped over the black dog

نمونه ورودی ۲:

CALLSECURITYATMIAMIAIRPORTBECAUSEITHINKABOMBISABOUTTOGOOFF

نمونه خروجی ۲:

calls security at miami airport because i think a bomb is about to go off