



گزارش پروژه عقب گرد

درس طراحی الگوریتم

مسئله «جیمز باند»

مجتبی فیاضی – 9728973

توضیحات کلی

هدف این مسئله پیدا کردن تمام جمله‌های ممکن از یک رشته سرهم با توجه به دیکشنری داده شده است. برای این منظور از تکنیک عقب‌گرد استفاده شده است. روال کار الگوریتم به این صورت است که هر یک از پیشوندهای جمله داده شده استخراج می‌شود و اگر امیدبخش بود (در دیکشنری موجود بود)، همین کار برای بقیه جمله بعد از این پیشوند تکرار می‌شود. بدین ترتیب تمام جملات ممکن از رشته مورد نظر استخراج می‌شود.

```
def breakTheSentence(string, n, result, dictionary):  
  
    for i in range(1, n + 1):  
        prefix = string[:i]  
        if wordIsInDictionary(prefix, dictionary):  
            if i == n:  
                # Add this element to previous prefix  
                result += prefix  
                print(result)  
                return  
  
            breakTheSentence(string[i:], n - i,  
                             result + prefix + " ", dictionary)
```

یک نمونه خروجی برنامه:

```
call security at miami air port be cause i think a bomb is about to go off  
call security at miami air port because i think a bomb is about to go off  
call security at miami airport be cause i think a bomb is about to go off  
call security at miami airport because i think a bomb is about to go off  
abort the plan meet at the dark cabin  
i love you so much
```

پیچیدگی زمانی

برای تخمین پیچیدگی زمانی الگوریتم، از روش مونته کارلو استفاده شده است. بدین منظور برای یک نمونه از مسئله، تابع estimate ده هزار بار فراخوانی شد و میانگین اینها به عنوان شاخص پیچیدگی گزارش شده است.

```
def estimate(sentence, dictionary):
    v = sentence
    numnodes, m, mprod = 1, 1, 1
    while m != 0:
        t = len(v)
        mprod = mprod * m
        numnodes = numnodes + mprod*t
        promisings = []
        for i in range(1, len(v) + 1):
            prefix = v[:i]
            if wordIsInDictionary(prefix, dictionary) and i != len(v):
                promisings.append(i)

        m = len(promisings)
        if m != 0:
            v = v[promisings[int(random.random() *
                                len(promisings))]:]

    return numnodes
```

نمونه تخمین پیچیدگی به ازای یک نمونه از مسئله:

```
call security at miami air port be cause i think a bomb is about to go off
call security at miami air port because i think a bomb is about to go off
call security at miami airport be cause i think a bomb is about to go off
call security at miami airport because i think a bomb is about to go off
estimate: 1392.9846
```