

[Algorithm] 2020 KAKAO BLIND RECRUITMENT - 문자열 압축 — 나무늘보의 개발 블로그

노트북: 첫 번째 노트북

만든 날짜: 2021-02-07 오후 11:51

URL: <https://continuous-development.tistory.com/229>

Algorithm

[Algorithm] 2020 KAKAO BLIND RECRUITMENT - 문자열 압축

2021. 2. 2. 22:43 수정 삭제 공개



kakao

문제 설명

데이터 처리 전문가가 되고 싶은 **어피치**는 문자열을 압축하는 방법에 대해 공부를 하고 있습니다. 최근에 대량의 데이터 처리를 위한 간단한 비손실 압축 방법에 대해 공부를 하고 있는데, 문자열에서 같은 값이 연속해서 나타나는 것을 그 문자의 개수와 반복되는 값으로 표현하여 더 짧은 문자열로 줄여서 표현하는 알고리즘을 공부하고 있습니다.

간단한 예로 aabbaccc의 경우 2a2ba3c(문자가 반복되지 않아 한번만 나타난 경우 1은 생략함)와 같이 표현할 수 있는데, 이러한 방식은 반복되는 문자가 적은 경우 압축률이 낮다는 단점이 있습니다. 예를 들면, abcabcbcdede와 같은 문자열은 전혀 압축되지 않습니다. 어피치는 이러한 단점을 해결하기 위해 문자열을 1개 이상의 단위로 잘라서 압축하여 더 짧은 문자열로 표현할 수 있는지 방법을 찾아보려고 합니다.

예를 들어, ababcbcdababcbcd의 경우 문자를 1개 단위로 자르면 전혀 압축되지 않지만, 2개 단위로 잘라서 압축한다면 2ab2cd2ab2cd로 표현할 수 있습니다. 다른 방법으로 8개 단위로 잘라서 압축한다면 2ababcbcdcd로 표현할 수 있으며, 이때가 가장 짧게 압축하여 표현할 수 있는 방법입니다.

다른 예로, abcabcbdede와 같은 경우, 문자를 2개 단위로 잘라서 압축하면 abcabc2de가 되지만, 3개 단위로 자른다면 2abcbdede가 되어 3개 단위가 가장 짧은 압축 방법이 됩니다. 이때 3개 단위로 자르고 마지막에 남은 문자열은 그대로 붙여주면 됩니다.

압축할 문자열 *s*가 매개변수로 주어질 때, 위에 설명한 방법으로 1개 이상 단위로 문자열을 잘라 압축하여 표현한 문자열 중 가장 짧은 것의 길이를 return 하도록 solution 함수를 완성해주세요.

제한사항

- *s*의 길이는 1 이상 1,000 이하입니다.
- *s*는 알파벳 소문자로만 이루어져 있습니다.

입출력 예

sresult

"aabbaccc"

"abababcdcdabababcdcd"	9
"abcababcdede"	8
"abcababcababcdededededede"	14
"xabababcdcdabababcdcd"	17

입출력 예에 대한 설명

입출력 예 #1

문자열을 1개 단위로 잘라 압축했을 때 가장 짧습니다.

입출력 예 #2

문자열을 8개 단위로 잘라 압축했을 때 가장 짧습니다.

입출력 예 #3

문자열을 3개 단위로 잘라 압축했을 때 가장 짧습니다.

입출력 예 #4

문자열을 2개 단위로 자르면 abcababcababc6de 가 됩니다.

문자열을 3개 단위로 자르면 4abcdededededede 가 됩니다.

문자열을 4개 단위로 자르면 abcababcababc3dede 가 됩니다.

문자열을 6개 단위로 자를 경우 2abcabc2dedede가 되며, 이때의 길이가 14로 가장 짧습니다.

입출력 예 #5

문자열은 제일 앞부터 정해진 길이만큼 잘라야 합니다.

따라서 주어진 문자열을 x / abababcdcd / abababcdcd 로 자르는 것은 불가능 합니다.

이 경우 어떻게 문자열을 잘라도 압축되지 않으므로 가장 짧은 길이는 17 이 됩니다.

문제 해설

문제에서는 문자열이 주어지고 반복되는 것에 따라 앞에 숫자를 붙여 중복을 줄일 수 있다. 이 때 가장 적은 길이의 문자열을 찾는 것이다. 나같은 경우에는 반복되는 모든 경우의 수를 만들고 그것들의 길이중에 가장 작은 값을 넣는 방식으로 했다.

```
def solution(s):
    list_numbers = [] # 모든 리스트를 넣어서 확인하기 위해 만들었다.
    answer=[] # answer 이다.
    if len(s)==1: # 길이 자체가 1 일 때는 바로 끝나게 했다. 테스트 케이스 5 번에 해당 된다.
        answer = len(s)
        return answer
    for i in range(1, round(len(s) / 2)+1): # 반복의 수는 전체 길이의 절반이상일 수 없어 값을 나눠서
        list_number = [] # 반복되는 개수가 1 일때, 2 일때 ... 이런식을 모두 세기 위해 만들었다.
        count = 1 # 2a 이런식으로 쓰일 때 앞자리에 들어가게 하기 위해 사용했다.
        for j in range(0, len(s), i): # 이 부분은 전체 문자열을 돌면서 반복되는 개수를 사용한다.
            if s[j:j+i] == s[j+i:j+i+i]: #이렇게 간격을 뒀다. i는 반복되는 개수로서 j부터 반복되는 횟수
                count += 1 # 앞의 값과 뒤에 값이 같으면 count를 하고 다시 for문으로 넘어간다.
            else:
                if count == 1: # 만약 같지 않다면 이제 count의 값에 따라서
                    list_number.append(s[j:j+i]) #list_number에 값을 그대로 넣는다.
                else: # count의 값이 1 이아니라니는 것은 반복이 된다는 의미이므로
                    list_number.append(str(count) + s[j:j+i]) #count를 앞으로 넣고 뒤에 반복되는 것을 넣어준다.
                    count = 1 # 그 후 count를 다시 1로 초기화 한다.
            else: # 이부분은 for문이 끝나면 오는 부분으로서
                if (len(s) - j)%i != 0: # 이 부분은 간격에 맞지 않아 들어가지 않는 부분을 넣어주기 위함
                    list_number.append(s[j+i:j+i+i]) # 맨 마지막에 오는 부분을 넣어준다.
                list_numbers.append(list_number) # 그 후 이 값을 list_numbers에 넣어준다.
        for i in range(0,len(list_numbers)): # 모든 반복에 대해 문자열을 변환하여 값을 구한뒤 이것을
            answer.append(len(''.join(list_numbers[i]))) # 하나씩 append를 한다. 그때 join문을 사용하여 리스트
        answer = min(answer) # 그 중 제일 짧은 값을 answer로 만든다.
    return answer
```

이 부분은 리스트컴프리션으로 만든 코딩이다. 파이썬에 최적화 되어 있는 부분이다.

```
# 파이썬 답게 짜놓은 코딩

def compress(text, tok_len):
    words = [text[i:i+tok_len] for i in range(0, len(text), tok_len)]
    res = []
    cur_word = words[0]
    cur_cnt = 1
    for a, b in zip(words, words[1:] + ['']):
```

```

    if a == b:
        cur_cnt += 1
    else:
        res.append([cur_word, cur_cnt])
        cur_word = b
        cur_cnt = 1
    return sum(len(word) + (len(str(cnt)) if cnt > 1 else 0) for word, cnt in res)

def solution(text):
    return min(compress(text, tok_len) for tok_len in list(range(1, int(len(text)/2) + 1)) + [len(text)])

a = [
    "aabbaccc",
    "abababcdcdabababcdcd",
    "abcababcdede",
    "abcabcbcabcdededededede",
    "xabababcdcdabababcdcd",

    'aaaaaa',
]

for x in a:
    print(solution(x))

```

'Algorithm' 카테고리의 다른 글

[Algorithm] 2020 KAKAO BLIND RECRUITMENT - 문자열 압축

[Algorithm] 2018 KAKAO BLIND RECRUITMENT - 다트게임

[Algorithm] 2018 KAKAO BLIND RECRUITMENT - 비밀 지도(문제 설명 및 문제...

[Algorithm] 2019 KAKAO 개발자 겨울 인턴쉽 - 크레인 뽑기(문제 설명 및 문제...

[Algorithm] 2019 KAKAO BLIND RECRUITMENT - 실패율 (문제 설명 및 문제 ...

[Algorithm] 40강 : 구간 합(Interval) 빠르게 구하기

문자열 압축

카카오

카카오 예제

나아무늘보

혼자 끄적끄적하는 블로그 입니다.

