# 이전 알고리즘

**answer 전역변수에 최종 답 저장**

**bool타입의 캐시 사용**

마지막에 선택한 문자열과 여태까지 선택한 문자열의 목록이 주어질 때, 동일한 형식을 이미 한번 체크했다면 더 이상 실행하지 않고 바로 리턴한다. 여기에서 문제가 발생하는데, 예를 들어 psa, augzq, y라는 문자열 조각이 있을 때, psa + augzq + y의 순서대로 조합하여 psaugzqy라는 문자열 합을 얻었다면, 이 다음에 augzq + psa + y의 순서대로 조합하려고 할 시 이를 실행하지 않고 바로 리턴해버린다는 것이다. 이는 메모이제이션도 아니며 함수 흐름을 방해할 뿐이다.

**last변수의 초기값이 0부터 시작**

처음 함수를 실행했을 때 각 문자열을 맨 앞에 오는 문자열로 한 번씩 선택하는 과정을 처리하는 부분을 참조자 ret 선언문의 아래부분에 두어 last 매개변수에 -1을 사용할 수 없었다. 따라서 last의 변수값이 벡터에 저장된 문자열 조각들의 인덱스보다 1이 크며, 이는 코드의 직관성을 해쳤다.

# 새 알고리즘

**restore 함수 자체가 최종 답 반환**

더 이상 전역변수 answer에 답을 저장할 필요가 없어졌다.

**string타입의 캐시 사용**

마지막에 선택한 문자열과 여태까지 선택한 문자열의 목록이 주어질 때, ‘마지막에 선택한 문자열’ + ‘남은 문자열 조각들을 합쳤을 때의 최소 길이 문자열’의 결과를 캐시에 저장한다. 이로써 이전 알고리즘에서 실행해야 할 루틴도 스킵해버리는 문제를 해결하였고, 메모이제이션도 잘 적용된다.

**last변수의 초기값이 -1부터 시작**

처음 함수 실행 시 last가 -1부터 시작하여 코드의 가독성을 높이고 좀 더 논리적으로 되었다.

**매개변수 curString의 제거**

애초에 함수가 재귀호출 되는 과정에서는 바로 이전에 선택한 문자열에 대한 정보만 알면 되며, 이는 매개변수 last로 대체할 수 있으므로 curString 매개변수를 제거했다. 이 알고리즘에서는 curString이 함수 내의 지역변수로써 사용되며, 바로 이전에 선택한 문자열 조각의 실질적인 문자열을 저장한다.