



# BOJ 12904 : A와 B

[통계](#) [수정](#) [삭제](#)

beberiche · 방금 전

0

[algorithms](#)[greedy](#)[백준](#)

## algorithms

1. BOJ 1593 : 문자해독
2. BOJ 11051 : 이항계수2
3. BOJ 2482 : 색상환
4. Programmers : 굴 고르기
5. BOJ 7569 : 토마토
6. BOJ 14500 : 테트로미오
7. BOJ 1504 : 특정한 최단 경로
8. Programmers :  $n^2$  배열자르기
9. BOJ 16947 : 서울 지하철 2호선
10. BOJ 16929 : Two Dots
11. BOJ 16719 : ZOAC
12. BOJ 1700 : 멀티탭 스케줄링
13. **BOJ 12904 : A와 B**

▲ 숨기기

13/13



BAE<JOON>  
ONLINE JUDGE

# 문제

12904번

제출

맞힌 사람

숏코딩

재채점 결과

채점 현황

내 제출

🔗 난이도 기여

강의▼

질문 게시판

A와 B

성공

☆

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	512 MB	9204	4322	3588	46.822%

## 문제

수빈이는 A와 B로만 이루어진 영어 단어가 존재한다는 사실에 놀랐다. 대표적인 예로 AB (Abdominal의 약자), BAA (양의 울음 소리), AA (용암의 종류), ABBA (스웨덴 팝 그룹)이 있다.

이런 사실에 놀란 수빈이는 간단한 게임을 만들기로 했다. 두 문자열 S와 T가 주어졌을 때, S를 T로 바꾸는 게임이다. 문자열을 바꿀 때는 다음과 같은 두 가지 연산만 가능하다.

- 문자열의 뒤에 A를 추가한다.
- 문자열을 뒤집고 뒤에 B를 추가한다.

주어진 조건을 이용해서 S를 T로 만들 수 있는지 없는지 알아내는 프로그램을 작성하시오.

## 입력

첫째 줄에 S가 둘째 줄에 T가 주어진다. ( $1 \leq S$ 의 길이  $\leq 999$ ,  $2 \leq T$ 의 길이  $\leq 1000$ , S의 길이 < T의 길이)

## 출력

S를 T로 바꿀 수 있으면 1을 없으면 0을 출력한다.

### 예제 입력 1 복사

```
B
ABBA
```

### 예제 출력 1 복사

```
1
```

### 예제 입력 2 복사

```
AB
ABB
```

### 예제 출력 2 복사

```
0
```

# 풀이과정

그리디 문제 한번 더.

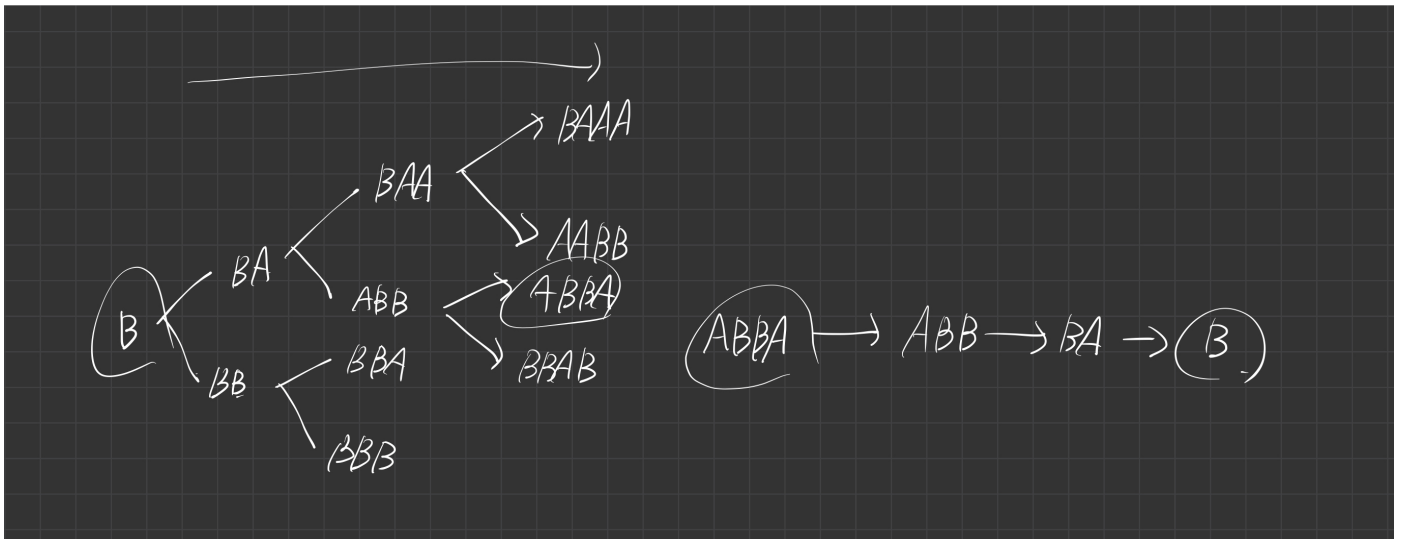
완전 탐색을 기반으로 하지만, 기존의 탐색 방법과는 다르게 접근해야한다.

시작부터 목표 언어까지 접근하게 되는 경우, 찾는 데 꽤 많은 메모리와 시간을 소모하여, 틀렸다는 답이 나오게 되더라.

그렇기 때문에, 시작과 목표점을 서로 바꾸어, 목표점에서 시작하여 변화시키는 과정 중에 똑같은 문자가 나오는지 확인하면 문제를 보다 효율적으로 풀 수 있다.

완전 탐색의 경우, BFS 를 적용하였다.

아마 이게 top-down , bottom-up 의 차이 인듯 한데, 요 원리가 그리디에 속하는지는 잘 모르겠다.



## 정답

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.util.LinkedList;
import java.util.Queue;

public class Main {
    static String strS;
    static String strT;

    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        strS = br.readLine();
        strT = br.readLine();

        solution(strT);

        System.out.println(0);
    }

    private static void solution(String st) {
        Queue<String> q = new LinkedList<>();
        q.add(st);

        while (!q.isEmpty()) {
            String curr = q.poll();
            if (curr.equals("")) continue;

            if (curr.equals(strS)) {
                System.out.println(1);
                System.exit(0);
            }

            if (curr.charAt(curr.length() - 1) == 'A') {
                q.add(curr.substring(0, curr.length() - 1));
            } else {
                StringBuilder sb = new StringBuilder();
                for (int i = curr.length() - 2; i >= 0; i--) {
                    sb.append(curr.charAt(i));
                }
                q.add(sb.toString());
            }
        }
    }
}
```



별

새로운 것에 관심이 많고, 프로젝트 설계 및 최적화를 좋아합니다.



이전 포스트

**BOJ 1700 : 멀티탭 스케줄링**

0개의 댓글

댓글을 작성하세요

댓글 작성