문자열(String)

문자열의 메소드

String str; str.equals(~~)

str.contains(~~)

str.charAt(i)

substring과 split

```
String str = "abcde";
String temp;
temp = str.substring(3,5) (3<=index<5)
temp <= "de"
String[] temp = new String[2];
temp = str.split("c");
temp = {ab,de};
```

StringBuilder의 StringBuffer

```
StringBuilder sb = new StringBuilder();

sb.append("ABC");

sb.reverse()

sb.charAt(0);

sb.deleteCharAt(0);

sb.equals("ABC")=?
```

StringBuffer vs StringBuilder (multithread)

kmp

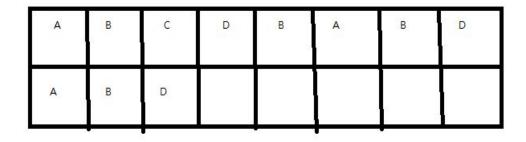
Knuth,Morris,Prett가 만든 문자열 탐색 알고리즘

길이N의 문자열안에 존재하는길이 M의 문자열을 찾을 때

단순 비교 시간복잡도 **O(NM)**

KMP 시간복잡도O(N+M)

단순 문자열 비교



| Α | В | С | D | В | A | В | D |
|---|---|---|---|---|---|---|---|
| | А | В | D | | | 1 | |

Kmp 알고리즘

문자열 M의 접두사, 접미사 확인

ABBAAB의 접두사

A AB ABB ABBA ABBAA ABBAAB

접미사

B AB AAB BAAB BBAAB ABBAAB

접두==접미인 부분 확인 (배열 선언)

ABBAAB

A arr[0] = 0

AB

ABB

ABBA

ABBAA

ABBA**AB** arr[5] = 2

| 인덱스 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|-----|---|---|---|---|---|---|---|---|---|---|----|----|
| 텍스트 | Α | В | С | D | Α | В | С | D | Α | В | E | Е |
| 패턴 | Α | В | С | D | Α | В | E | | | | | |

| 인덱스 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|-----|---|---|---|---|---|---|---|
| 텍스트 | Α | В | С | D | А | В | С |
| 패턴 | Α | В | С | D | А | В | E |

| 인덱스 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|---|---|---|---|-------|---|---|---|---|---|----|
| 텍스트 | Α | В | С | D | A (i) | В | С | D | Α | В | E |
| 패턴 | | | | | A (j) | В | С | D | А | В | E |