

C++ String 시간복잡도

| | |
|--|-------------------------------------|
| ≡ 태그 | |
| <input checked="" type="checkbox"/> 공개여부 | <input checked="" type="checkbox"/> |
| <input type="text"/> 날짜 | |
| <input type="text"/> 작성일자 | |

[1번 케이스]

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string s;
    int n = 1000000;
    for (int i = 0; i < n; i++) {
        s += "A";
    }
    return 0;
}
```

이것은 $O(N)$ 의 복잡도가 걸린다

[2번 케이스]

```
#include <iostream>#include <string>using namespace std;

int main() {
    string s;
    int n = 1000000;
    for (int i = 0; i < n; i++) {
        s = s + "A";
    }
    return 0;
}
```

이것은 $O(N^2)$ 의 복잡도가 걸린다

1번 케이스는 문자열 s의 마지막에 "A"가 추가되는 방식으로 돌아간다.

2번 케이스는 매번 새로운 임시 문자열을 만든다고 할 수 있다.

즉, 문자열 A와 B가 있다고 하면, S문자열의 크기 + 더할 문자열의 크기의 합만큼 연산을 해야 한다.

새로운 임시 문자열을 만드는 것의 시간복잡도(최악의 경우) = $O(N)$

$O(N)$ 연산을 n 번하는 꼴이고, $O(N^2)$ 의 시간 복잡도를 가진다.

```
import java.util.*;
public class Main {
    public static void main(String args[]){
        String s = "";
        int n = 10000;
        for(int i=0;i<n;i++){
            s += "A";
        }
    }
}
```

여담으로 자바의 경우에는 $s += "A"$ 이라고 써도 $s = s + "A"$ 로 변환되기 때문에 항상 $O(N^2)$ 이 걸린다.

정리

- C++에서 string 의 += 연산은 $O(K)$ 이다.
- C++에서 string 의 + 연산은 $O(N+K)$ 이다.

For **built-in types** `a += b` is exactly the same as `a = a + b` but for classes, those operators are overloaded and call different functions.

In your example `fans = fans + s[i]` creates a temporary string, and assigns (moves) it to `fans`, but `fans += s[i]` does not create that temporary, hence it may be faster.