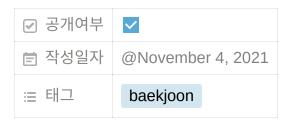


## [Baekjoon] 11723 - 집합



문제풀이를 다시 시작하면서, 알고리즘도 한번 싹 훑으려고 하던 중 비트마스크가 눈에들어왔다. 물론 비트마스크는 알고리즘이 아닌 테크닉이지만, 문제풀이에 있어서 실행시간을 줄여주는 테크닉이기때문에, 오랜만에 다시한번 문제를 풀어보았다.

```
#include <iostream>
#include <string>

using namespace std;

int main(void)
{
    ios::sync_with_stdio(false); cin.tie(NULL); cout.tie(NULL);

    int M;
    int S = 0;
    string buf;

    cin >> M;
    for (int i = 0; i < M; i++)
    {
}</pre>
```

[Baekjoon] 11723 - 집합 1

```
int X;
        bool tmp;
        cin >> buf;
        if (buf.compare("all") == 0) {
            S = (1 << 21) - 1;
            continue;
        }
        else if (buf.compare("empty") == 0) {
            S = 0;
            continue;
        }
        cin >> X;
        switch (buf[0])
        case 'a':
            S = S \mid (1 << X);
            break;
        case 'r':
            S = S & \sim (1 << X);
            break;
        case 'c':
            tmp = S & (1 << X);
            cout << tmp << '\n';
            break;
        case 't':
            S = S \wedge (1 << X);
            break;
        }
    }
    return 0;
}
```

비트마스킹이 디버깅하기가 조금 애매하다보니까, 개인적으로는 조금 많이 복잡하게 느껴진다. 그래도 연습하다보면 조금 적응될것같긴하다. 이전에는 비트마스킹을 사용하지 않고 풀었는데, 그때의 시간과 비교해보니 그닥 많은 차이가 나는것같지는 않다. (오히려 메모리를 더 사용하는 것같기도...)

[Baekjoon] 11723 - 집합 2