



[Baekjoon] 1654 - 랜선 자르기

☰ 태그	baekjoon
☑ 공개여부	☑
📅 작성일자	@January 20, 2022

나무자르기에서 한번 더 엮은? 조금 더 심화한 문제라고 보면 된다. 나무자르기의 경우 원하는 나무만큼을 얻고, 최대의 절단기 높이를 구했다면 이번엔 원하는 랜선개수만큼을 구하는데, 길이가 최대일때를 구함과 동시에, 한번만 자르면 되었던 나무자르기완 다르게 자를수 있는만큼 계속 자르면 되었다.

```
#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

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

    long long K, N, result, L, R, mid;
    vector<long long> cable;

    cin >> K >> N;
    for(int i = 0; i < K; i++) {
        long long tmp;
        cin >> tmp;
```

```

        cable.push_back(tmp);
    }

    sort(cable.begin(), cable.end(), greater<long long>());
    L = 0;
    R = cable[0];
    result = 0;
    while(L <= R)
    {
        int cnt = 0;

        mid = (L + R) / 2;
        for(int i = 0; i < cable.size(); i++)
            for(int h = 1; mid * h <= cable[i]; h++)
                cnt++;

        if(cnt >= N) {
            if(mid > result)
                result = mid;
            L = mid + 1;
        }
        else
            R = mid - 1;
    }

    cout << result << '\n';

    return 0;
}

```

맞는거같긴한데, 채점 80%에서 시간초과가 뜬다. 이 말은 내가보기엔 아무래도 개수를 구하는 과정에서 나는 시간초과로 보았다.

어쨌든 Cnt가 N보다 크기만하면 되고, 우리가 원하는건 최대 길이이기때문에, 정확한 Cnt는 구할 필요가 없었다. 그래서 for문에 $cnt \leq N$ 을 추가하였다. 바로 정답이 뜨는걸 볼 수 있었다.

```

#include <iostream>
#include <vector>
#include <algorithm>

using namespace std;

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

    long long K, N, result, L, R, mid;
    vector<long long> cable;

```

```

cin >> K >> N;
for(int i = 0; i < K; i++) {
    long long tmp;
    cin >> tmp;
    cable.push_back(tmp);
}

sort(cable.begin(), cable.end(), greater<long long>());
L = 0;
R = cable[0];
result = 0;
while(L <= R)
{
    int cnt = 0;

    mid = (L + R) / 2;
    for(int i = 0; i < cable.size(); i++)
        for(int h = 1; mid * h <= cable[i] && cnt <= N; h++)
            cnt++;

    if(cnt >= N) {
        if(mid > result)
            result = mid;
        L = mid + 1;
    }
    else
        R = mid - 1;
}

cout << result << '\n';

return 0;
}

```