

# 2805\_나무자르기

정동욱

# 문제 설명

- 몇 미터의 높이에서 나무를 비 가장 낭비를 덜하는지 구하는 문제이다.

# 문제 해결

- $1m \sim Xm$ (가장 긴 나무의 높이) 중 가장 낭비를 하지않는 높이를 찾아야하므로 이분 탐색을 이용하면 빠른 시간안에 찾을수 있다.

# 핵심 코드

```
24     scanf("%lld %lld",&n,&m);
25
26     for(long long i=0;i<n;++i){
27         long long temp;
28         scanf("%lld",&temp);
29         tree.push_back(temp);
30     }
31     sort(tree.begin(),tree.end()); //나무의 높이를 정렬
32
33     long long left = 0; // 0으로 초기화
34     long long right = tree.back(); // 가장 긴 나무의 높이로 초기화
35     long long ans=0;
36     while(left<=right){ // 이분탐색을 통해 값을 찾는다.
37         long long mid = (left+right)/2;
38         if(check(mid)){
39             ans=max(ans,mid);
40             left=mid+1;
41         }
42         else{
43             right=mid-1;
44         }
45     }
46     printf("%lld\n",ans);
47
```

# 시간 복잡도

- 나무의 높이를 정렬하는데  $O(n \cdot \log(n))$  이 소요되고 이진탐색을 통해 답을 찾을때 최악의 경우  $O(\log(n))$  이 소요되므로 시간 복잡도는  $O(n \cdot \log(n))$  이다.