

## 문제정의

주어진 일을 끝낼 수 있는 최소한의 불도저 개수를 구하는 문제

## 문제해결

### 입력

시작날을 기준으로 작업을 저장할 1차원 배열 `job`을 선언한다(초기값은 전부 0). `si`를 입력 받을 때 `job[si]`의 값을 1씩 증가한다.

### 이진 탐색

불도저의 개수를 `b`라고 할 때, 불도저의 개수를 이진탐색을 통해 지정하여 주어진 일을 끝낼 수 있는지 검사한다. 이진탐색의 `left`와 `right`값은 1과 `m`으로 초기화한다.

`b`를  $(left+right)/2$ (이하 `mid`)로 하여 이진탐색을 통해 `mid`개의 불도저가 주어진 일을 끝낼 수 있는지 검사한다. 정답을 저장하는 `res`는  $10^9$ 로 초기화한다.

`left`가 `right`보다 작을 동안 다음을 반복한다.

주어진 일을 끝낼 수 있을 경우

정답을 저장하는 변수 `res`에  $\min(res, mid)$ 를 저장하고, 더 적은 불도저수로 일을 끝낼 수 있는지 확인하기 위해 `right`에 `mid`를 대입한다.

주어진 일을 끝낼 수 없을 경우

더 많은 불도저가 필요하므로 `left = mid+1`로 갱신한다.

`mid` 값을 갱신한다.

### 검사

주어진 일을 끝낼 수 있는 지 확인하기 위해 변수를 추가하였다. 현재 불도저가 하고 있는 일의 시작날짜를 `late`라 하고 불도저가 한 일들을 저장하는 1차원 배열 `complete`를 생성하였다. 초기 `complete`의 원소는 모두 0이다.

`i`번째 날짜( $1 \leq i \leq n$ )에 대해 다음을 반복한다.

`late+d < i`인 경우(불도저가 마감안에 일을 처리하지 못한 경우)

`false`를 반환한다.

`else`(마감기한 내에서 일을 하고 있는 경우)

불도저의 수만큼 `complete[late]`를 증가시킨다.

`late`에 해당하는 일을 다했으나 `i`째날의 일이 남았을 경우

`late`를 1증가하고 남은 불도저의 수만큼 `complete[late]`를 증가한다.

`late`에 해당하는 일을 다하고 `i`번째 일도 했을 경우

`i`를 1증가하여 다음날로 넘어간다.

반복이 끝났을 때 모든 날짜에 대해서 `job[i]==complete[i]`라면 모든 일을 끝냈으므로 `true`를 반환하고 그렇지 않은 경우 `false`를 반환한다.

## 시간복잡도

$1 \sim m$ 사이의 값을 이진탐색하므로  $\log M$ 의 시간이 소요되고 선택된 `mid`값에 대하여 최대 될 수 있는 `mid`는  $(M/2)+1$ 이므로 검사에서 소요되는 최대 시간 복잡도는  $N \times (M/2)+1$ 이다. 따라서  $O(NM \log M)$ 이다.

## 공간복잡도

`complete`와 `job`모두 최대 `n`인 100000으로 선언하였으므로, 공간복잡도는  $O(1)$ 이다.

## 토의

에러케이스를 찾지 못하여 검사부분에서의 오류를 찾지 못하였다.