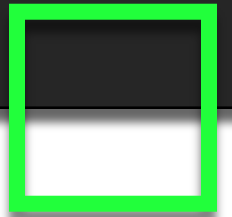


#Algorithm 반  
#1주차

# Binary Search, Parametric Search

T. 이제인  
Asst. 최창규 강건





# 시작하기 전에...

BOJ 백준 온라인 저지 가입!

[www.acmicpc.net](http://www.acmicpc.net) / [boj.kr](http://boj.kr)

→ '2019 ALOHA' 그룹 가입!

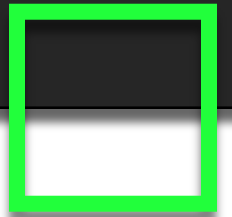
[www.acmicpc.net/group/5031](http://www.acmicpc.net/group/5031)



# 목차

- 0. 알고리즘 Intro
- 1. Binary Search
- 2. Parametric Search

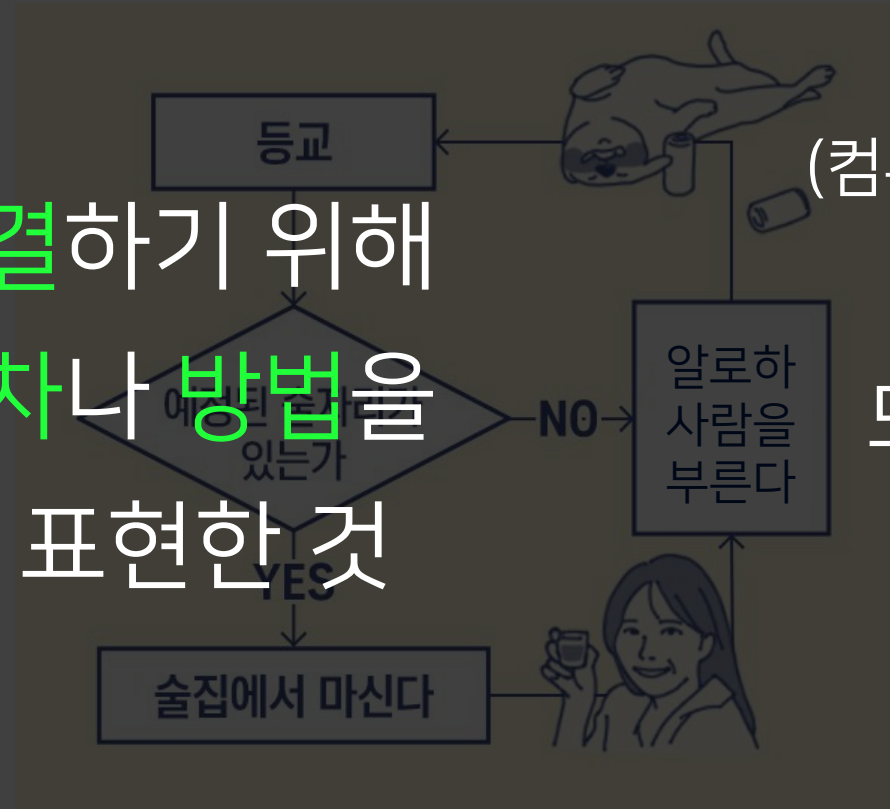
# Algorithm 알고리즘?





# 알고리즘이란?

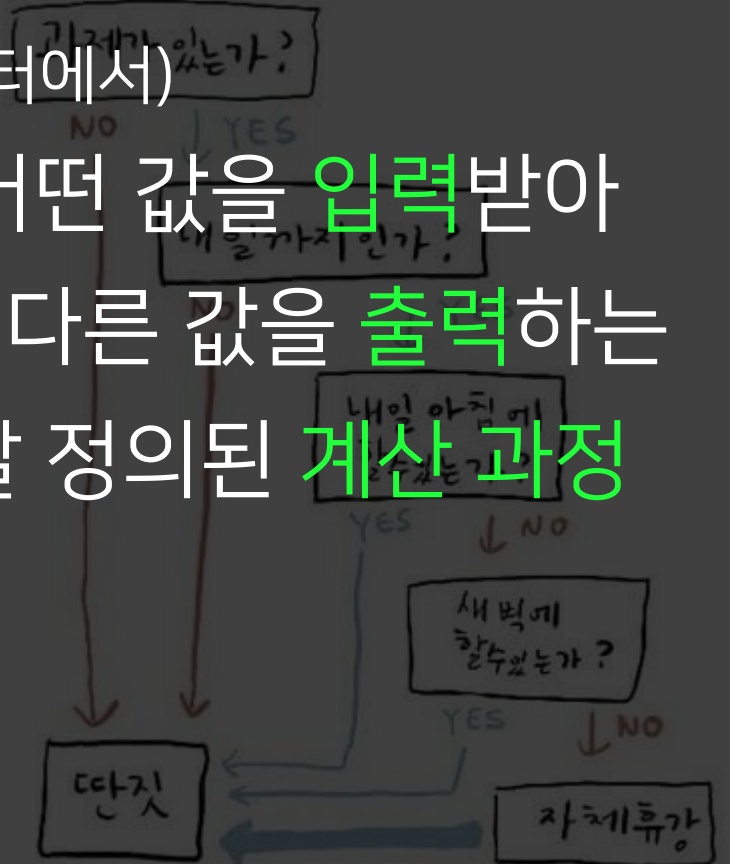
어떠한 문제를 해결하기 위해  
정해진 일련의 절차나 방법을  
공식화한 형태로 표현한 것



과제 알고리즘

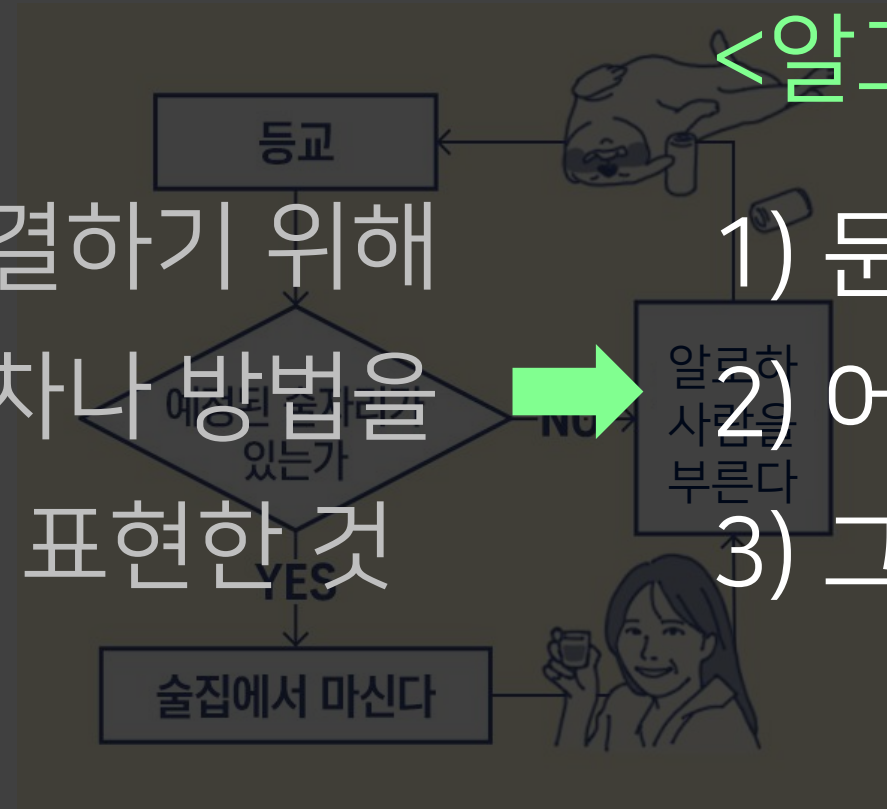
(컴퓨터에서)

어떤 값을 입력받아  
또 다른 값을 출력하는  
잘 정의된 계산 과정



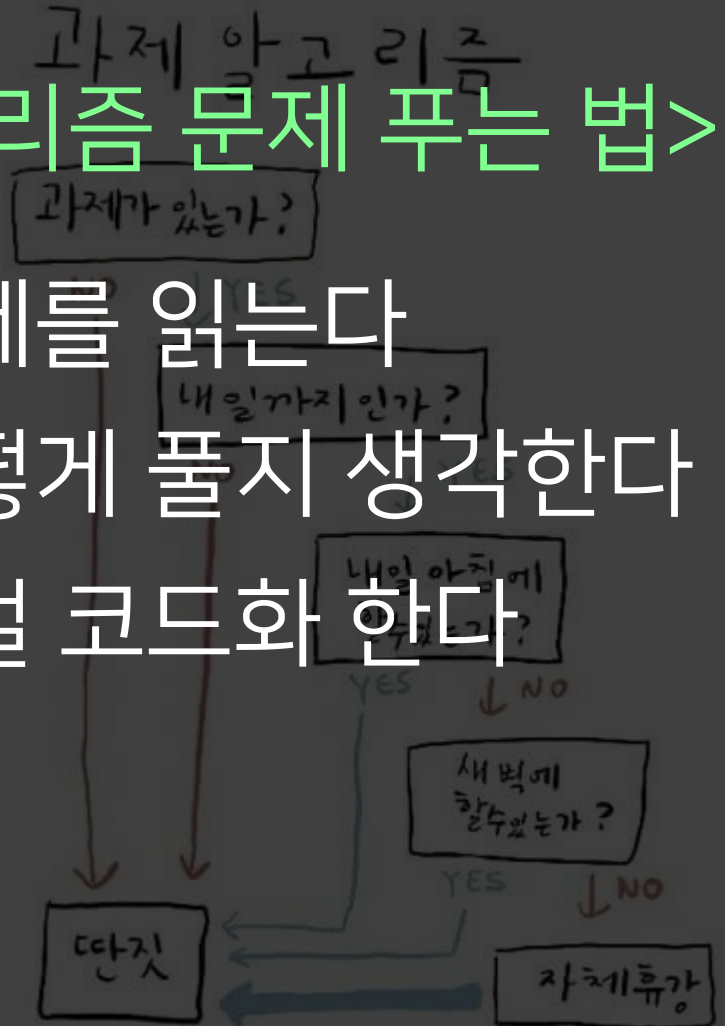
# 알고리즘이란?

어떠한 문제를 해결하기 위해  
정해진 일련의 절차나 방법을  
공식화한 형태로 표현한 것



## <알고리즘 문제 푸는 법>

- 1) 문제를 읽는다
- 2) 어떻게 풀지 생각한다
- 3) 그걸 코드화 한다

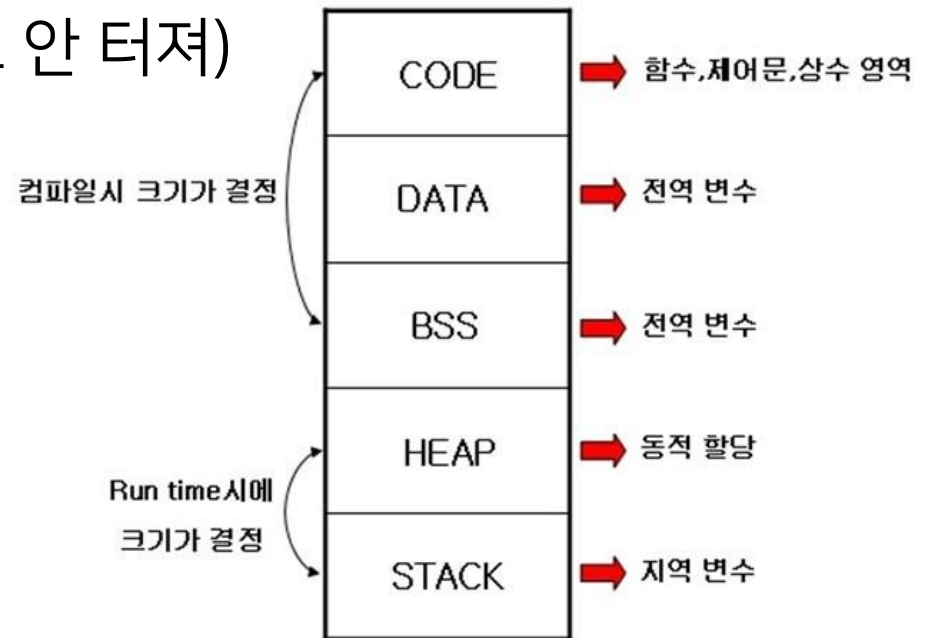




# 알고리즘 문제 풀 때 알아야할 것 - PS (problem solving)

- 함수를 잘 쓰자  
(똑같은 코드 또 쓰지 않게)
- 읽기 좋은 코드를 쓰자  
코딩은 이쁘게!
- 좋은 건 배우자  
(남의 코드 베껴도 괜찮아)
- 입력 개수 보고 시간 복잡도 생각  
→ 그 후에 문제 풀기

- 전역변수를 쓰자 (웬만하면)  
(크기 커도 안 터져)



- 동적할당 (malloc), 포인터 고민 L L  
(강 다 선언해 버려)



# 알고리즘 문제 풀 때 꼭 알아야할 것 - 자료구조

자료구조 (Data Structure) 란?

- 컴퓨터에서 자료를 효율적으로 관리하고 구조화시키는 방법
- 효율적으로 문제를 처리하려면 자료를 보관, 정리하는 기술이 필요!
- 컴퓨터는 자료의 특성에 따라 다양한 자료구조 기법을 사용

# 알고리즘 문제 풀 때 필요한 자료구조

자료구조	선언 방법 예시	특별히 필요한 헤더	특징
배열	<code>int arr[50];</code>	X	딱히??..... 자주 씬 ㅇㅇ
벡터	<code>std::vector&lt;int&gt; v;</code>	<code>#include &lt;vector&gt;</code>	중간에 끼워 넣고 빼는 거 가능. 미리 크기 안 정해놔도 됨
스택	<code>std::stack&lt;int&gt; st;</code>	<code>#include &lt;stack&gt;</code>	LIFO. 괄호 문제 풀 때 씬
큐	<code>std::queue&lt;int&gt; qu;</code>	<code>#include &lt;queue&gt;</code>	FIFO. BFS할 때 씬
우선순위큐	<code>std::priority_queue&lt;int&gt; pq;</code>	<code>#include &lt;queue&gt;</code>	트리 구조. 큰 값이 위로 간다 (max heap이랑 똑같다) 위상정렬할 때 씬
맵	<code>std::map&lt;int, char&gt; m;</code>	<code>#include &lt;map&gt;</code>	(key, value) 꼴. 넣으면 자기 스스로 정렬해줌 (key값으로 value값 찾기 같은 느낌) (개빠름)

(지금 다 몰라도 나중에 공부할 때 참고하세요!!)

# 알고리즘 문제 풀 때 필요한 것 - #include<algorithm>

The algorithms library defines functions for a variety of purposes (e.g. searching, sorting, counting, manipulating) that operate on ranges of elements. (출처 : cppreference)

>> 해석 : 문제 풀 때 개꿀이다!

\* 써먹기 좋은 예시들 (써먹기 좋다는 기준은 내 마음)

<code>max(a, b) / min(a, b)</code>	a, b 중에 큰 / 작은 값 리턴
<code>lower_bound(arr, arr+n, a)</code> <code>lower_bound(vec.begin(), vec.end(), a)</code>	정렬된 배열/벡터 안에서 a가 어느 위치에 들어가야할 지 (a보다 작은 값, 큰 값의 경계) 이터레이터 리턴
<code>next_permutation(vec.begin(), vec.end())</code>	{1,2,3,4}면 1234,1243,1324,1342.... 이렇게 다음 순열로 바꿈 (바꿀 수 있을 때까지 true 리턴)

(애도 지금은 몰라도 나중에 잘 써먹어보세요)



이 헤더에 뭐가 더 있는지 궁금하면

>> <https://en.cppreference.com/w/cpp/header/algorithm>

# 그 중에 우리가 오늘 쓸 것 - Sort(정렬)

다양한 정렬 방법 있다! (자료구조론, 알고리즘 수업 때 배울 거야)

[www.toptal.com/developers/sorting-algorithms](http://www.toptal.com/developers/sorting-algorithms) (멍때릴 때 보기 좋은 움짤)

그 중에 가장 빠르다고 할 수 있는 방법 == 퀵소트 (quick sort)

→ 그 방법으로 구현 된 함수가 알고리즘헤더에 있다!!!

→ 우리는 그냥 갖다 쓰면 된다 (자기가 직접 구현해도 되지만 문제 풀 땐 그러지 말자)

# 그 중에 우리가 오늘 쓸 것 - Sort(정렬)

쓰는 방법

배열 : `sort(arr, arr+n)` / 벡터 : `sort(v.begin(), v.end())`

(`arr[0]~arr[n-1]`를 정렬하고 싶을 때)

\*\*\* `using namespace std;` 까먹지 말자!!

↑ ↑ 이대로 쓰면 오름차순(1234.....)로 정렬됨.

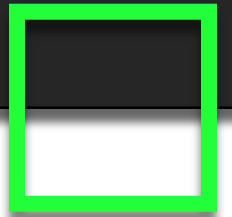
나는 다른 방법으로 하고 싶은데? 할 땐 다른 함수 넣자

ex) 내림차순(9876...): `sort(arr, arr+n, desc)`

`bool desc(int a, int b) { return a>b; }`

(아니면 `#include <functional> greater<int>()` 써도 된다!)

# Binary Search



# Binary Search (이분탐색)



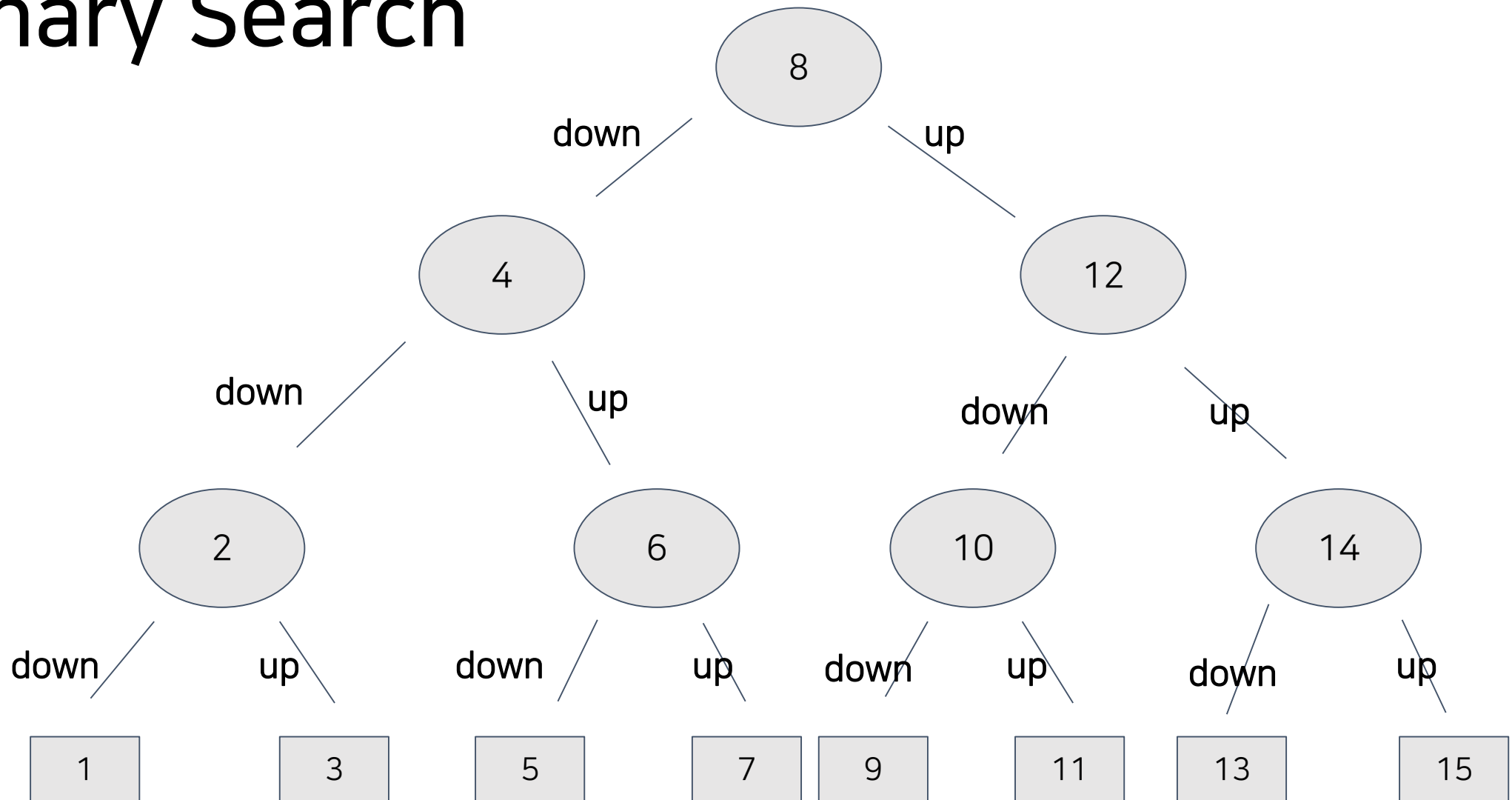
1~50 사이의 숫자 중  
이 안에 쓰여진 수를 맞춰라! (Up/Down)

→ 효율적으로 게임을 하려면  
맨 처음에 불러야 하는 수는? == 25

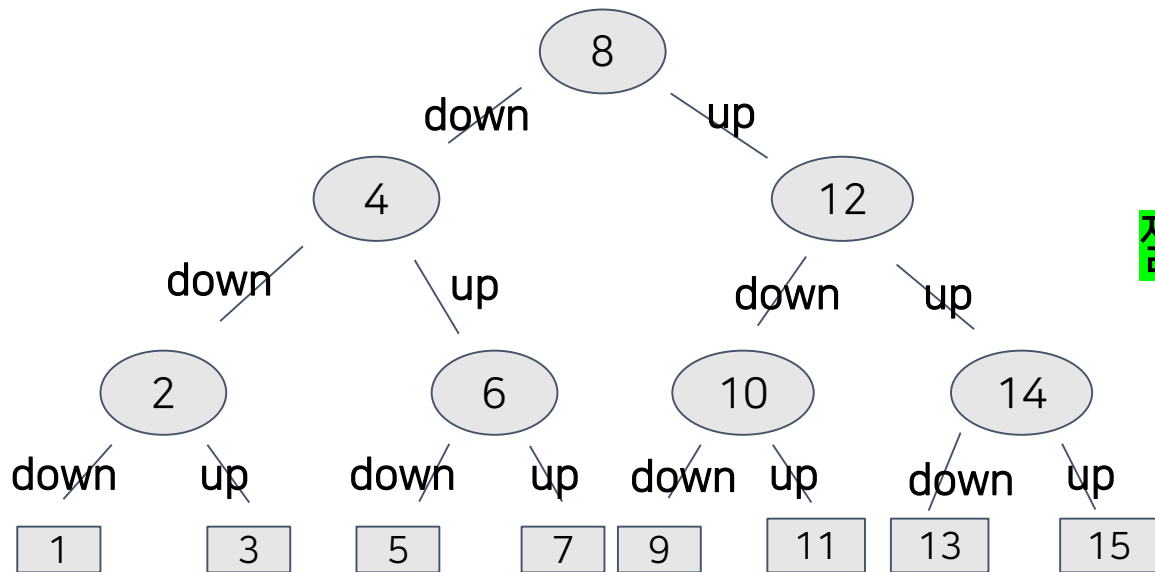




# Binary Search

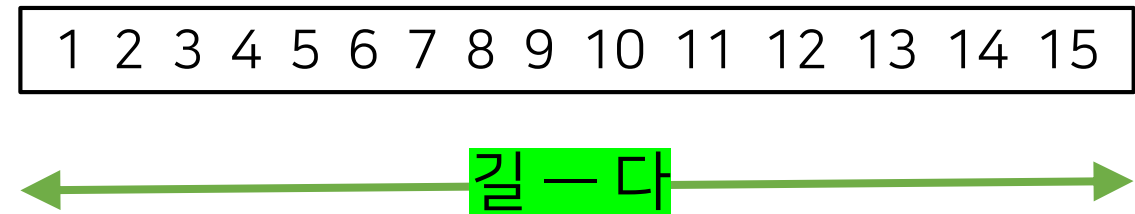


# Binary Search



최대  $\log_2 n$  번 비교  
 $O(\lg n)$

vs Linear Search (선형탐색)



vs

최대  $n$  번 비교  
 $O(n)$

# Binary Search 알고리즘

1. arr 정렬 (오름차순)
2. left==0, right==n-1 부터 시작
3. mid 정하기
4. value가 arr[mid]보다 크다 → mid+1 ~ right 확인 (3번부터 반복)  
value가 arr[mid]보다 작다 → left ~ mid-1 확인 (3번부터 반복)  
value랑 arr[mid]랑 같다 → 찾았다!! return 하기 (끝내기)

\*\* left, right, mid 모두 인덱스!!! 값 아님!!!

# Binary Search 알고리즘

4.  $\text{value} > \text{arr}[\text{mid}] \rightarrow \text{left} = \text{mid} + 1$ , 반복  
 $\text{value} < \text{arr}[\text{mid}] \rightarrow \text{right} = \text{mid} - 1$ , 반복  
 $\text{value} == \text{arr}[\text{mid}] \rightarrow \text{return}$

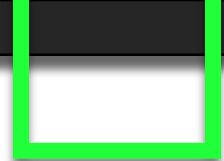
value가 arr에 없으면  
무한루프?????

# Binary Search 알고리즘

4.  $\text{value} > \text{arr}[\text{mid}] \rightarrow \text{left} = \text{mid} + 1$ , 반복  
 $\text{value} < \text{arr}[\text{mid}] \rightarrow \text{right} = \text{mid} - 1$ , 반복  
 $\text{value} == \text{arr}[\text{mid}] \rightarrow \text{return}$

value가 arr에 없으면  
무한루프?????

→  $\text{left} \leq \text{right}$  일 때만 반복 ( $\text{left} > \text{right}$  면 없다고 리턴해버려)



# 연습 문제



BOJ 1920  
수 찾기

# Binary Search 코드

```
int main() {
    int N, M, num;
    scanf("%d", &N);
    for (int i = 0; i < N; i++) scanf("%d", &A[i]);
```

배열 A를 정렬

```
    scanf("%d", &M);
    while (M--) {
        scanf("%d", &num);
        printf("%d\n", BS( ));
    }
    return 0;
}
```

재귀함수

```
int BS(int left, int right, int num) {
    int mid = ;
    if (left <= right) {
        if ( )
            return BS( , , num);
        else if ( )
            return BS( , , num);
        else return 1; // 찾았다!!
    }
    return 0; // 값 존재 X
}
```

반복문

```
int BS(int left, int right, int num) {
    int mid;
    while (left <= right) {
        mid = ;
        if ( )
            right = ;
        else if ( )
            left = ;
        else return 1; // 찾았다!!
    }
    return 0; // 값 존재 X
}
```





# 연습 문제



BOJ 10815  
숫자 카드

BOJ 10816  
숫자 카드 2

# 과제

(앞에 거부터 먼저 풀기!!)

+ 다른 방법으로 풀 수 있어도 바이너리 서치로 다시 풀어보세요!!



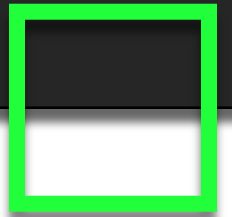
BOJ 2869

달팽이는 올라가고 싶다

BOJ 2792

보석 상자

# Parametric Search



# Parametric Search

매개변수

16. 좌표평면 위를 움직이는 점  $P$  의 시각  $t$  ( $0 < t < \pi$ )에서의 위치  $P(x, y)$ 가

$$x = \sqrt{3} \sin t, \quad y = 2 \cos t - 5$$

이다. 시각  $t = \alpha$  ( $0 < \alpha < \pi$ )에서 점  $P$ 의 속도  $\vec{v}$ 와  $\overrightarrow{OP}$ 가 서로 평행할 때,  $\cos \alpha$ 의 값은? (단,  $O$ 는 원점이다.) [4점]

①  $\frac{1}{10}$

②  $\frac{1}{5}$

③  $\frac{3}{10}$

④  $\frac{2}{5}$

⑤  $\frac{1}{2}$



# Parametric Search

바이너리 서치 : mid에 있는 값이 답인지?

파라메트릭 서치 : 이 값이 답의 범위에 만족하는지? → 범위를 줄여가면서 답 찾기

ex) 1번~10번 사람이 학점 순으로 정렬되어 있다.

장학금을 받을 수 있는 가장 턱걸이의 사람의 번호는?

# Parametric Search

바이너리 서치 : mid에 있는 값이 답인지?

파라메트릭 서치 : 이 값이 답의 범위에 만족하는지? → 범위를 줄여가면서 답 찾기

ex) 1번~10번 사람이 학점 순으로 정렬되어 있다.

장학금을 받을 수 있는 가장 턱걸이의 사람의 번호는?

→ 3.5 넘나요? 라고 물어 봤을 때

1) 넵ㅎㅎ : 정답 범위는 맞는데 턱걸이를 찾는 거니까 나보다 위 번호들 ㅎㅎ

2) 아녀ㅠ : 나도 못 받으니까 내 밑에는 당연히 못 받으니 아래 번호들 ㅎㅎ



# 연습 문제



BOJ 1654  
랜선 자르기



# BOJ 1654 랜선 자르기

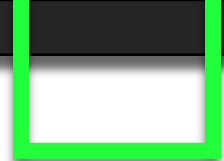
랜선의 길이는  $2^{31}-1$ 보다 작거나 같은 자연수이다.

→

```
#include <climits>
// 정수 최댓값(INT_MAX) 불러오는 헤더
// 아니면 #define INF 987654321 하기

int main() {
    ...
    PS(1, INT_MAX);
    ...
}
```

```
long long PS(long long left, long long right) {
    ...
    long long mid = (left + right) / 2;
    if (left <= right) {
        for (int i = 0; i < K; i++)
            [redacted];
        if ([redacted])
            return PS([redacted], [redacted]);
        else if ([redacted])
            return PS([redacted], [redacted]);
    }
    ...
}
```



# 연습 문제



BOJ 2512  
예산

# BOJ 2512 예산

n개의 수의 합이 m보다 클 때

→ mid보다 작은 쪽 : arr 값 그냥 더하기  
큰 쪽 : mid 값 더하기

→ 합이 m보다 크면  $right = mid - 1$  해서 반복  
작거나 같으면  $left = mid + 1$  해서 반복 && 답 갱신  
(가능한 거 중에 최댓값 찾는 거니까)

# BOJ 2512 예산

```
// 그냥 그대로 ㅇㅋ
if (sum <= m) printf("%d", );

// 이분탐색해서 찾기
else {
    int left = ; int right = ; int ans = 0;
    while (left <= right) {
        int mid = (left + right) / 2;
        sum = 0;
        ;

        if ( ) right = mid - 1;
        else {
            left = mid + 1;
            ans = ; // 최대니까 갱신 해주기
        }
    }
    printf("%d", ans);
}
```

# 연습 문제



BOJ 2805  
나무 자르기

# BOJ 2805 나무 자르기

랜선 자르기 : (랜선 길이 / mid) 다 더한 값 비교

나무 자르기 : (나무 높이 - mid) 다 더한 값 비교

(나머진 다 똑같다)

\*팁 : long long 주의

# 연습 문제



BOJ 2613  
숫자구슬





# 과제

(앞에 거부터 먼저 풀기!!)

+ 다른 방법으로 풀 수 있어도 바이너리 서치로 다시 풀어보세요!!



BOJ 3079  
입국 심사

BOJ 1300  
K번째 수



ALPHA  
The algorithm club.

# 고생했어요



ALPHA  
The algorithm club.