

이분(진) 탐색

백석대학교 강윤희



차례

- 문제정의
- 알고리즘 (반복)
- 알고리즘 (재귀)
- 분석



문제 정의

- 크기 순서대로 정렬된 리스트(자료)에서 특정 값이 있는지 찾아 그 위치를 반환하고자 함, 리스트에 없으면 -1을 반환함
- 이분 탐색(binary search) 는 이분, 둘로 나눈다는 의미임
 - 탐색할 자료를 둘로 나누어 찾아감
 - 탐색 영역을 줄여가며 찾음
- (사례) 사전에서 단어 찾기, 호텔에서 방의 호수 찾기



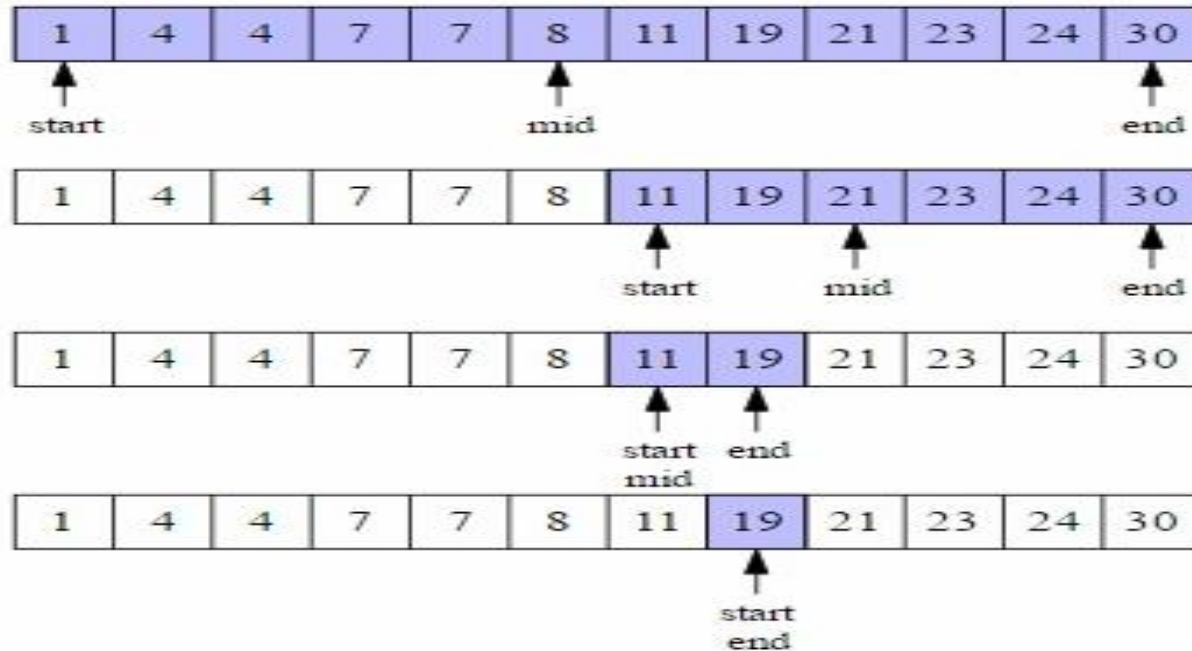
문제 정의

- 크기 순서대로 정렬된 리스트(자료)에서 특정 값이 있는지 찾아 그 위치를 반환하고자 함, 리스트에 없으면 -1을 반환함
- (작성 내용)알고리즘의 동작을 이해하기 위해 자료를 찾으면 True 반환, 자료를 못찾으면 False 을 반환



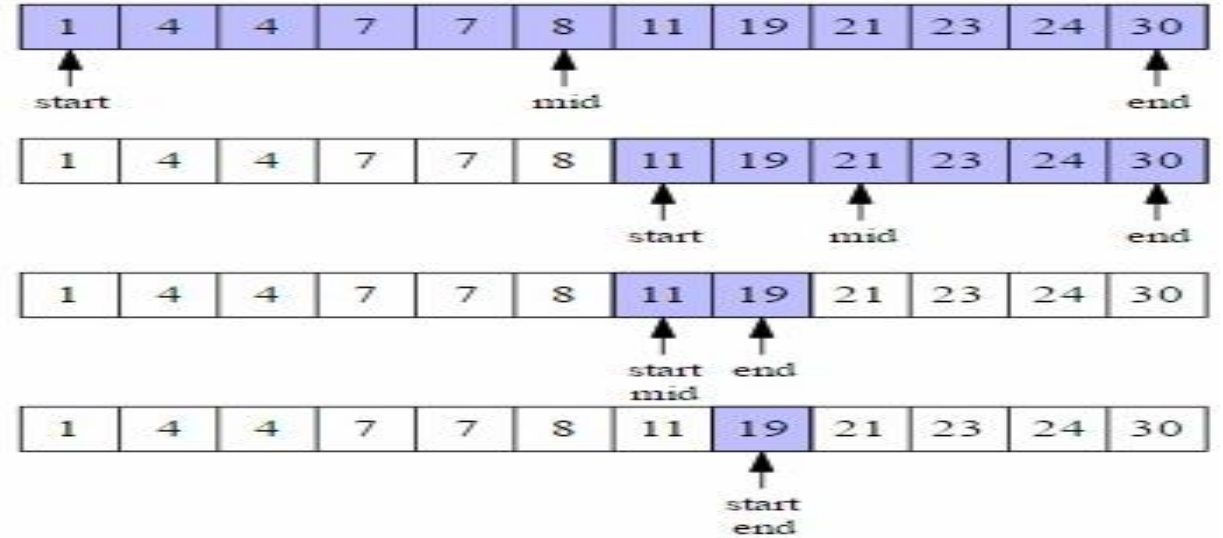
알고리즘 작성

```
def binarySearch(arr, item):  
    first = 0  
    last = len(arr)-1  
    found = False  
    while first <= last and not found:  
        midpoint = (first + last)//2  
        if arr[midpoint] == item:  
            found = True  
        else:  
            if item < arr[midpoint]:  
                last = midpoint-1  
            else:  
                first = midpoint+1  
  
    return found
```



알고리즘 작성(재귀)

```
def binarySearch(arr, item):  
    if len(arr) == 0:  
        return False  
    else:  
        midpoint = len(arr)//2  
        if arr[midpoint] == item:  
            return True  
        else:  
            if item < arr[midpoint]: # midpoint 보다 작은곳에서 찾음  
                return binarySearch(arr[:midpoint], item)  
            else:  
                return binarySearch(arr[midpoint+1:], item)
```



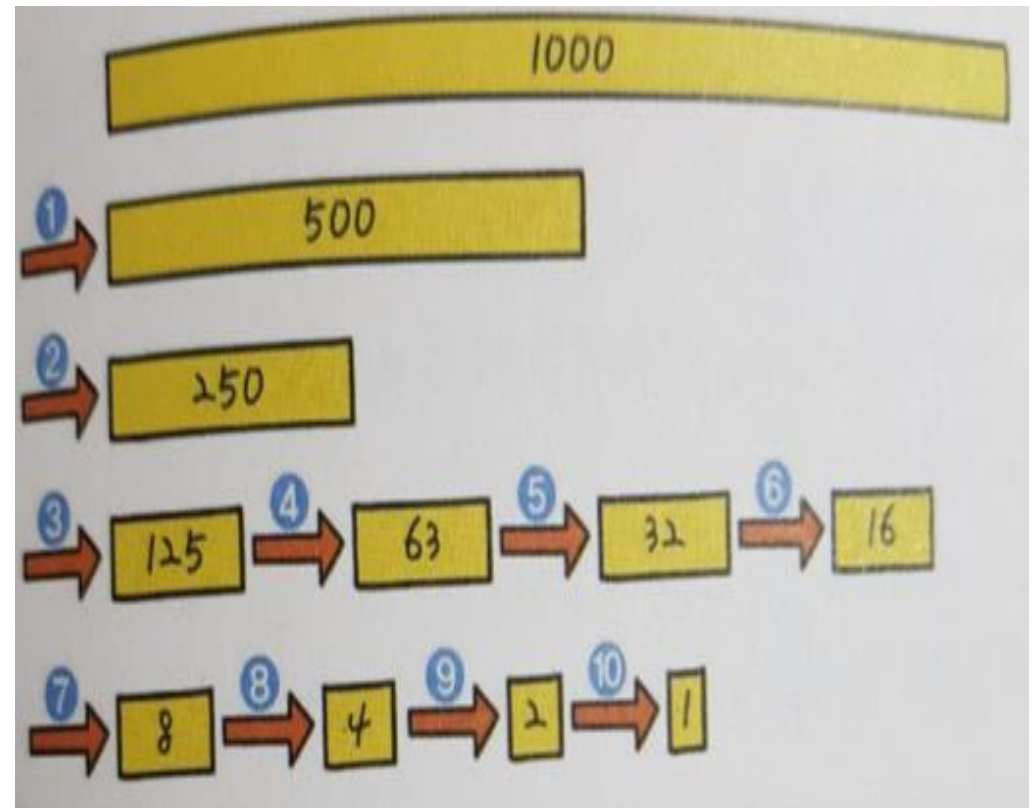
분석

- 값을 비교하여 찾는 값이 있을 범위를 절반씩 좁히면서 탐색
- 선형탐색에서 비해 효율적인 탐색 알고리즘임
- 천개의 자료에서 원하는 자료 찾는 경우 최악의 경우 자료 열개와 비교하여 찾을 수 있음 (밑이 2인 로그계산기 사용)

$$\log_2 1000 (9.966) < \log_2 1024 (10)$$

- 계산 복잡도는 $O(\log n)$, 순차탐색 $O(n)$ 보다 효율적임
- 국민 오천만 명 중에서 특정 주민번호를 찾고자 함, 최악의 경우 몇번에 찾을 수 있는가 ?

$$\log_2 50,000,000 (25.575)$$



분석

이진 로그 (binary logarithm)는 밑이 2인 로그
 \log_2 로 표기하며, 2의 거듭제곱의 역함수

$$x = \log_2 n \iff 2^x = n. \quad n \text{은 자료수, } x \text{는 비교횟수}$$

밑이 2인 로그 구하기:

Log_2 = 9.965784284662087

밑이 2인 로그 구하기:

Log_2 = 25.5754247590989

참고

<http://mwultong.blogspot.com/2008/01/2-log-calc.html>

