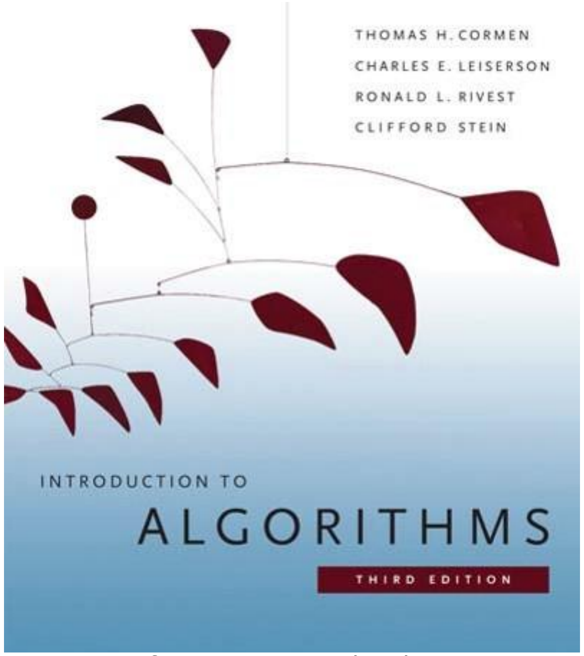
6.006- 알고리즘 개론



3강

목차

- 정렬!
  - 삽입 정렬
  - 합병 정렬
- 재귀 풀이

정렬 문제

입력: 숫자 배열 A[1…n]

출력: A 의 순서 배열 B[1…n]
예시: $B[1] \le B[2] \le \ldots \le B[n]$.

예시: A = [7, 2, 5, 5, 9.6] → B = [2, 5, 5, 7, 9.6]

어떻게 하면 효과적으로 정렬할 수 있을까?

왜 정렬을 해야 할까?

- 확실한 응용 사례
  - MP3 보관함 정렬
  - 전화번호부 정리
- 일단 정렬되면 쉬워지는 문제들
  - 중간값 또는 가장 가까운 쌍 찾기
  - 이진 탐색, 통계적 이상치 확인
- 생소한 응용 사례
  - 데이터 압축: 정렬로 중복된 부분 검출
  - 컴퓨터 그래픽: 앞에서 뒤로 화면 렌더링

삽입 정렬

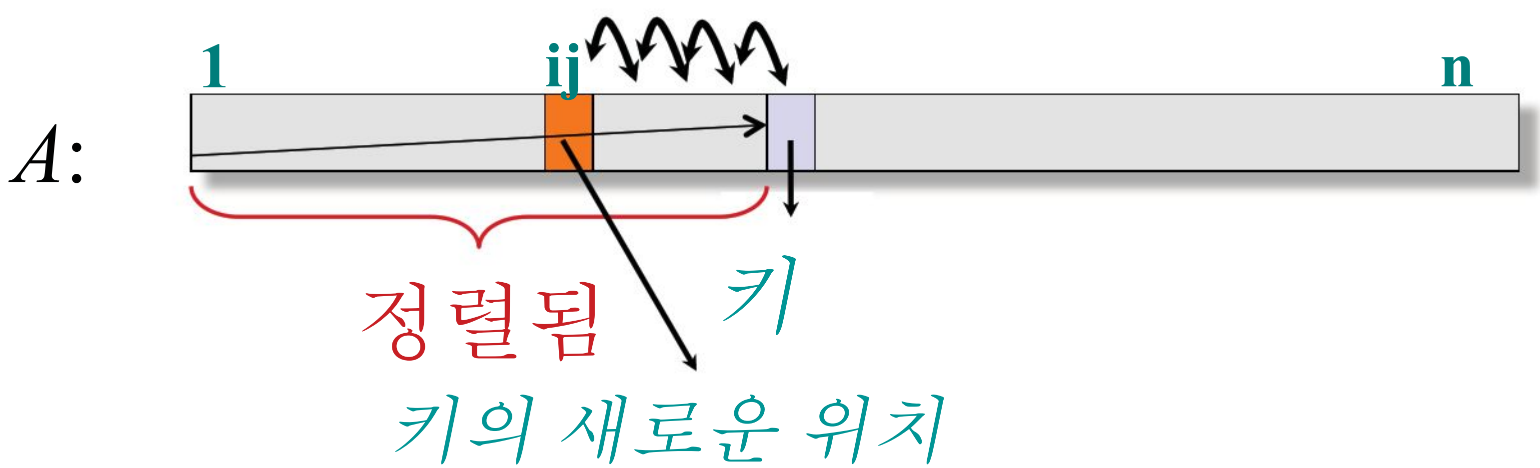
삽입 정렬 (A, n) ▷ $A[1 \,..\, n]$

$j \leftarrow 2$ 부터 n 까지 반복

알맞은 자리로 쌍별 스왑을 통해서

키 $A[j]$ 를 (이미 정렬된) 보조 배열인 $A[1 \,..\, j\text{-}1]$ 에 삽입

반복되는 j에 대한 그림 :



삽입 정렬의 예시

8 2 4 9 3 6

삽입 정렬의 예시

8 2 4 9 3 6

삽입 정렬의 예시

8 2 4 9 3 6

2 8 4 9 3 6

삽입 정렬의 예시

8 2 4 9 3 6

2 8 4 9 3 6

삽입 정렬의 예시

8 2 4 9 3 6

2 8 4 9 3 6

2 4 8 9 3 6

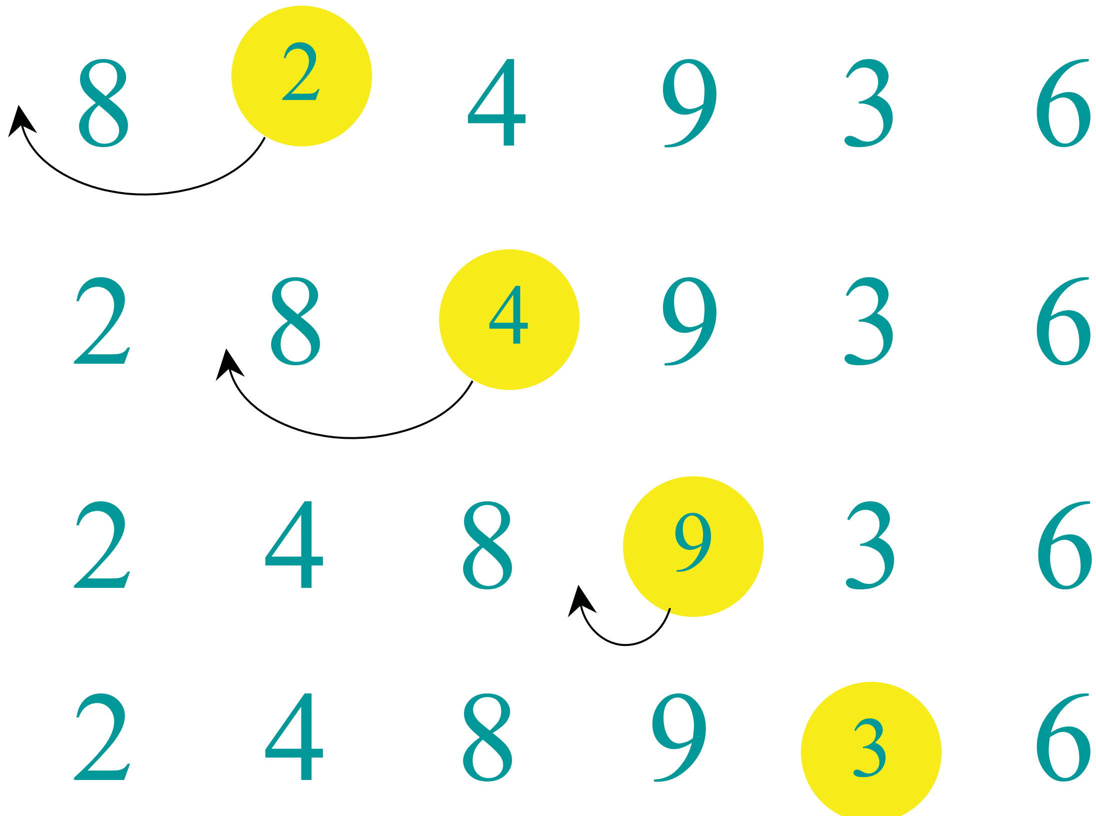
삽입 정렬의 예시

8 2 4 9 3 6

2 8 4 9 3 6

2 4 8 9 3 6

삽입 정렬의 예시



삽입 정렬의 예시

8 2 4 9 3 6

2 8 4 9 3 6

2 4 8 9 3 6

2 4 8 9 3 6

삽입 정렬의 예시

8 2 4 9 3 6

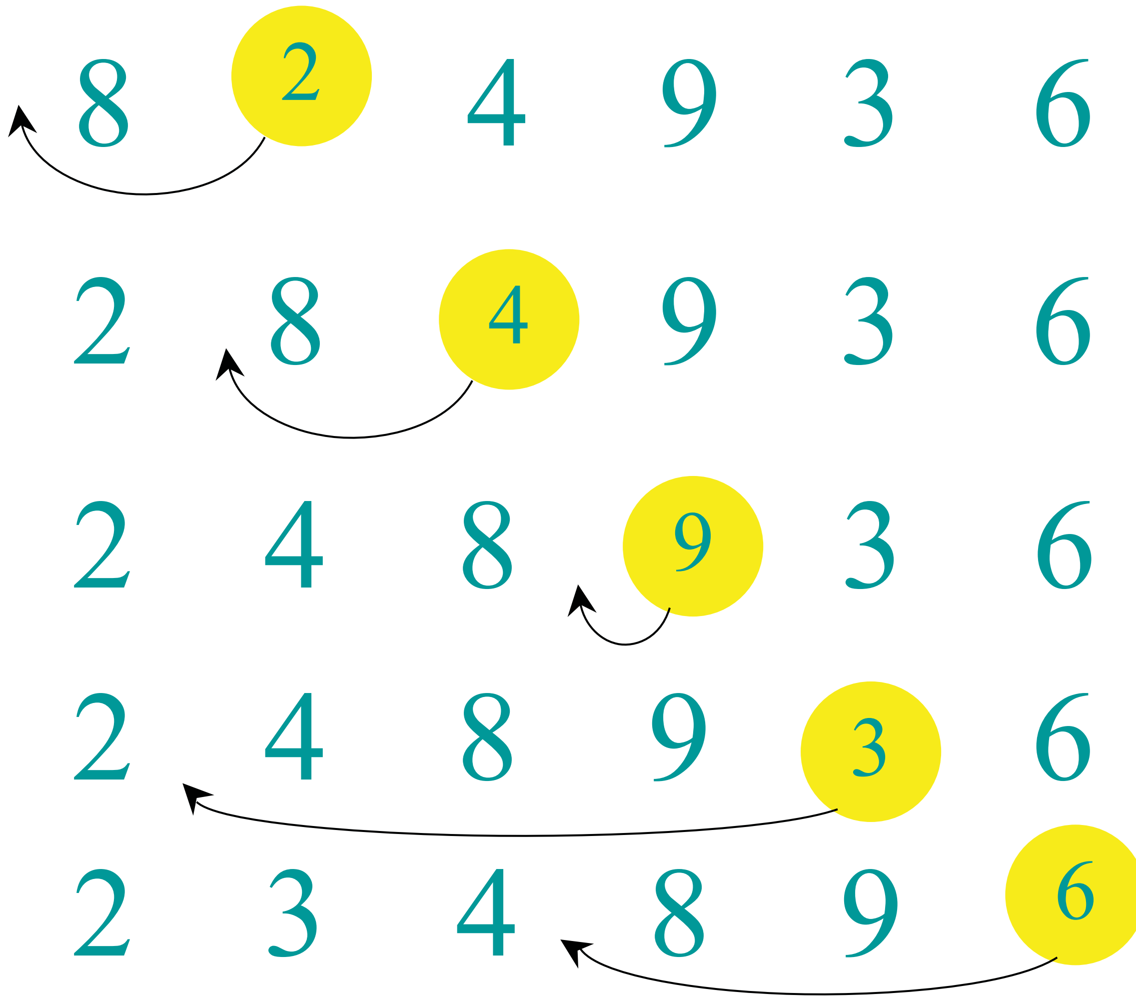
2 8 4 9 3 6

2 4 8 9 3 6

2 4 8 9 3 6

2 3 4 8 9 6

삽입 정렬의 예시



삽입 정렬의 예시

8 2 4 9 3 6

2 8 4 9 3 6

2 4 8 9 3 6

2 4 8 9 3 6

2 3 4 6 8 9 완료

실행 시간은? $\Theta(n^2)$임. 왜냐하면 $\Theta(n^2)$ 비교와 $\Theta(n^2)$ 스왑이므로.
예시: 입력값이 $A = [n, n-1, n-2, \ldots, 2, 1]$인 경우

이진 삽입 정렬

이진 삽입 정렬 (A, n) $\triangleright$ $A[1\,.\,.\,n]$

$j \leftarrow 2$ **부터** n **까지 반복**

(이미 정렬된) 보조 배열인 A[1 .. j-1]**에 키** A[j] **삽입.**

알맞은 위치를 찾기 위해 이진 탐색 사용.

이진 탐색은 Θ(log n) 시간이 걸림.

하지만 삽입 후 항목을 옮기는 것도 Θ(n) 시간이 걸림.

복잡도: Θ(n log n) 비교

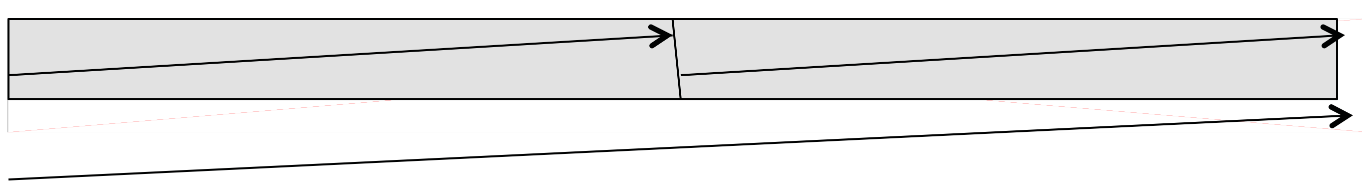
(n^2) 스왑

합병 정렬이란?

분할 정복

합병 정렬 $A[1\,..\,n]$

1. $n = 1$일 때, 바로 끝 (정렬할 거 없음).
2. 그 외의 경우, $A[\,1\,..\,n/2\,]$과 $A[\,n/2+1\,..\,n\,]$을 재귀적으로 정렬.
3. 두 정렬된 보조 배열 "**합병**"



핵심 서브 루틴: 합병

정렬된 두 배열 합병

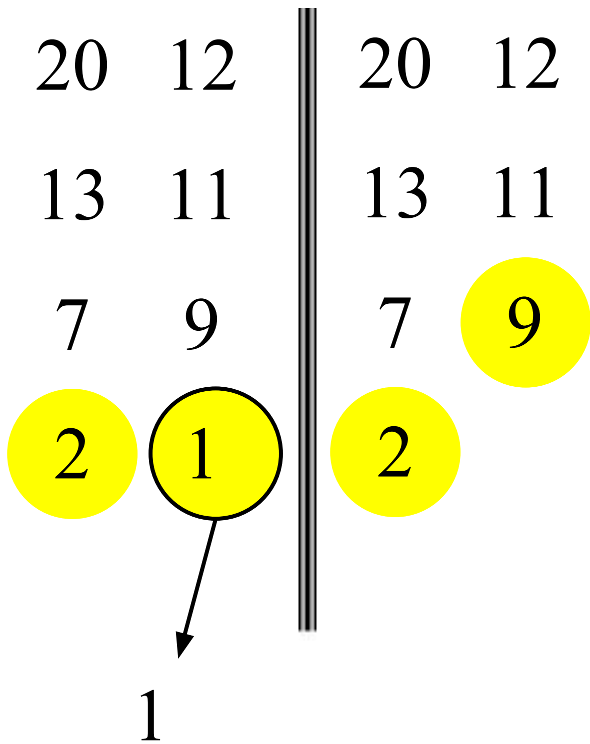
20	12
13	11
7	9
2	1

정렬된 두 배열 합병

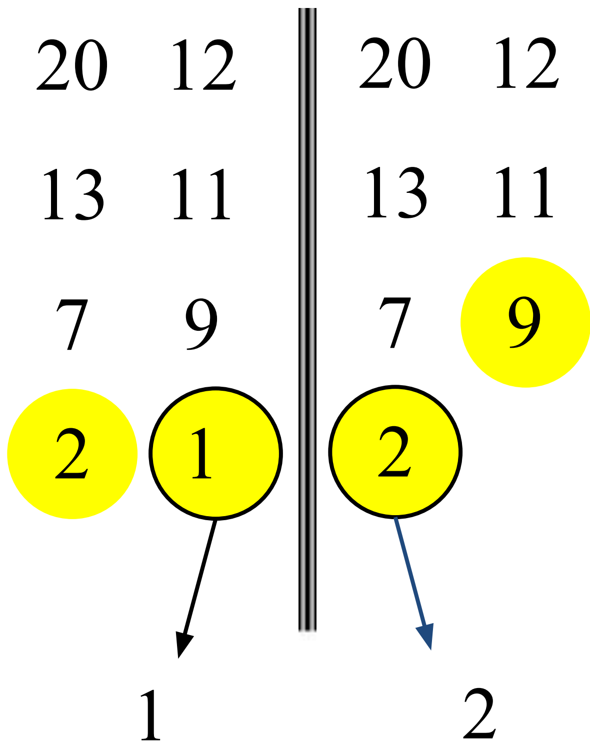
20	12
13	11
7	9
2	1

1

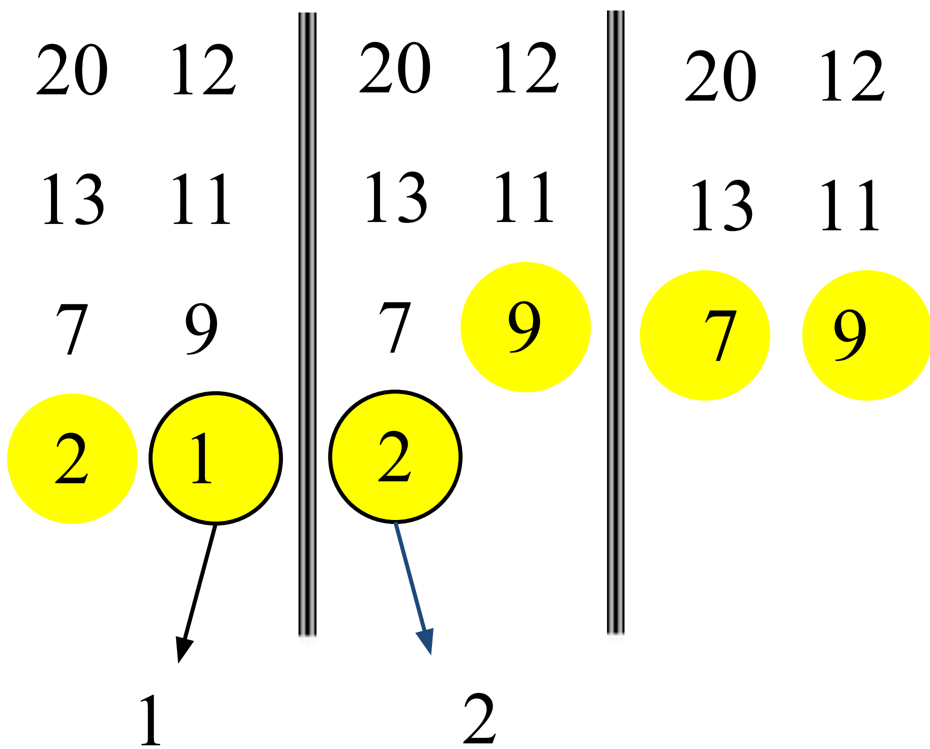
정렬된 두 배열 합병



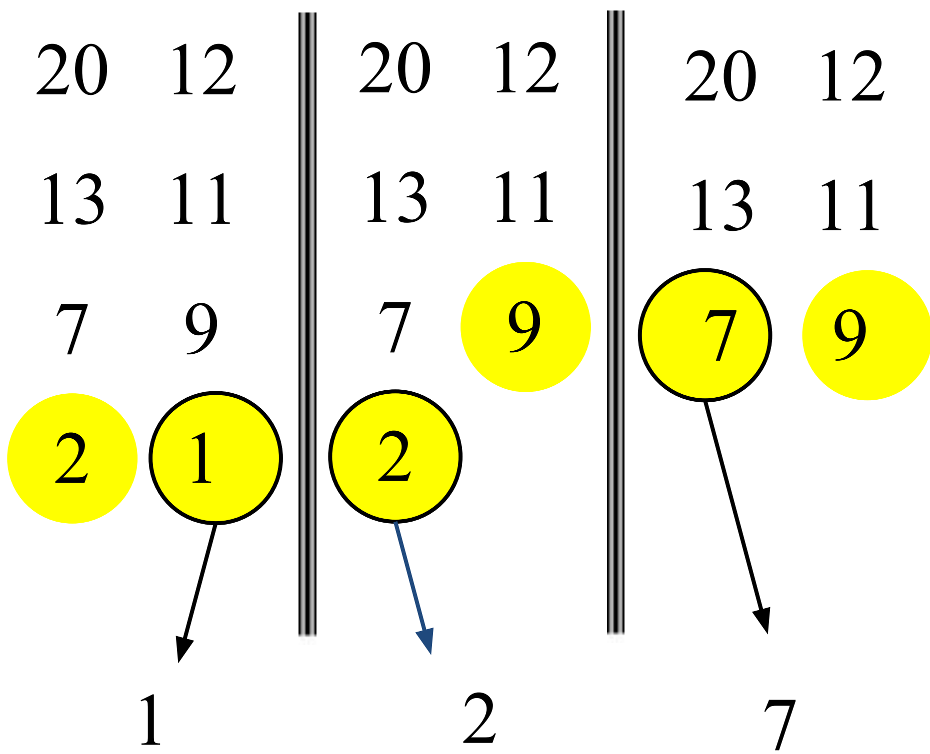
정렬된 두 배열 합병



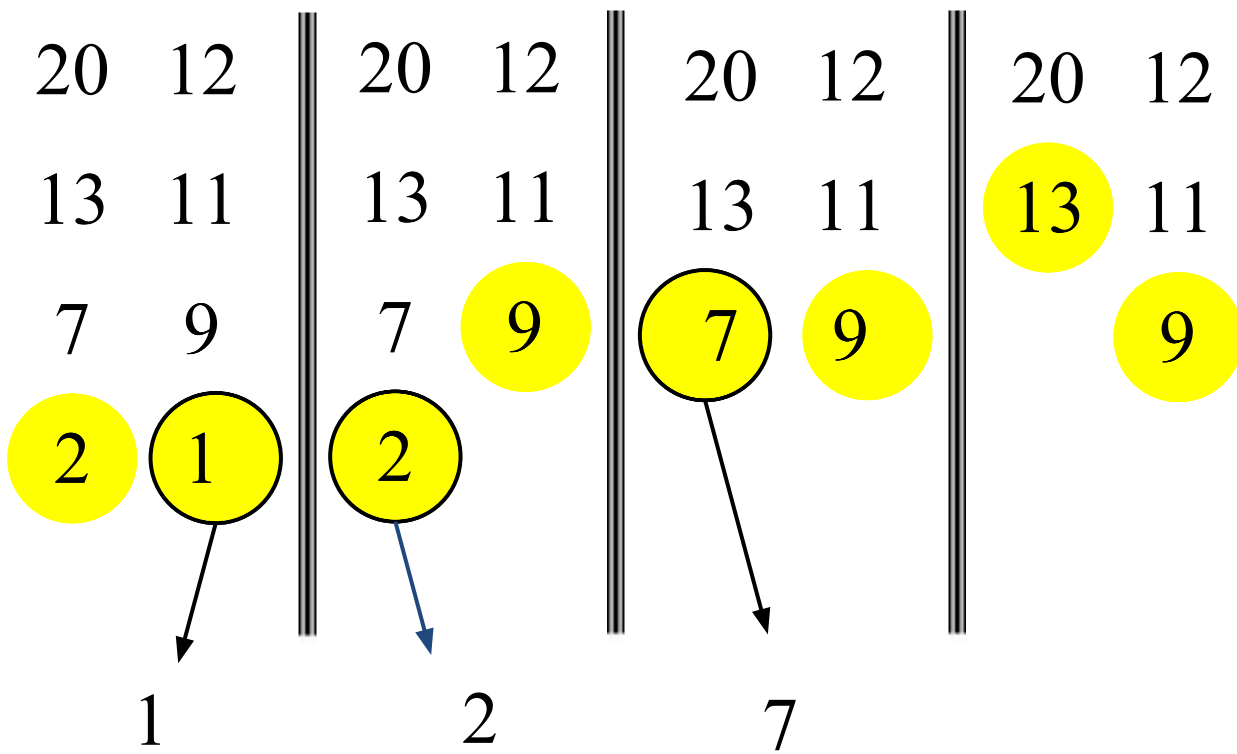
정렬된 두 배열 합병



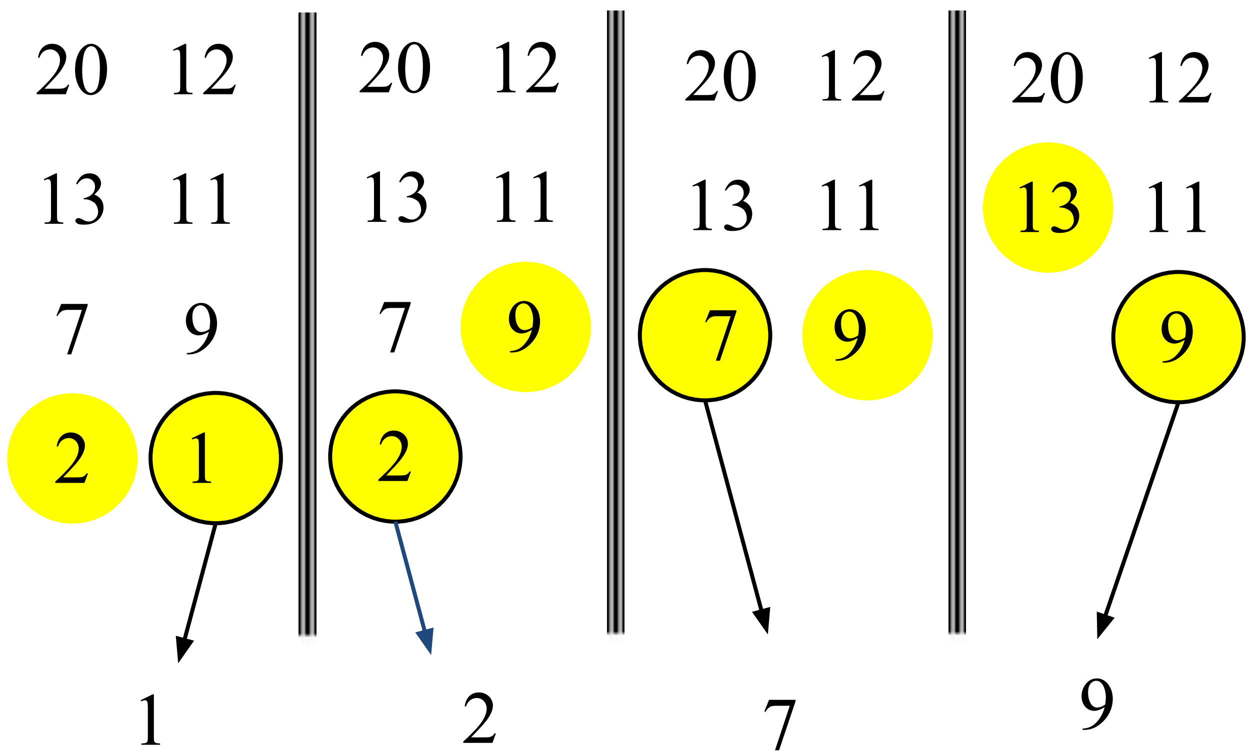
정렬된 두 배열 합병



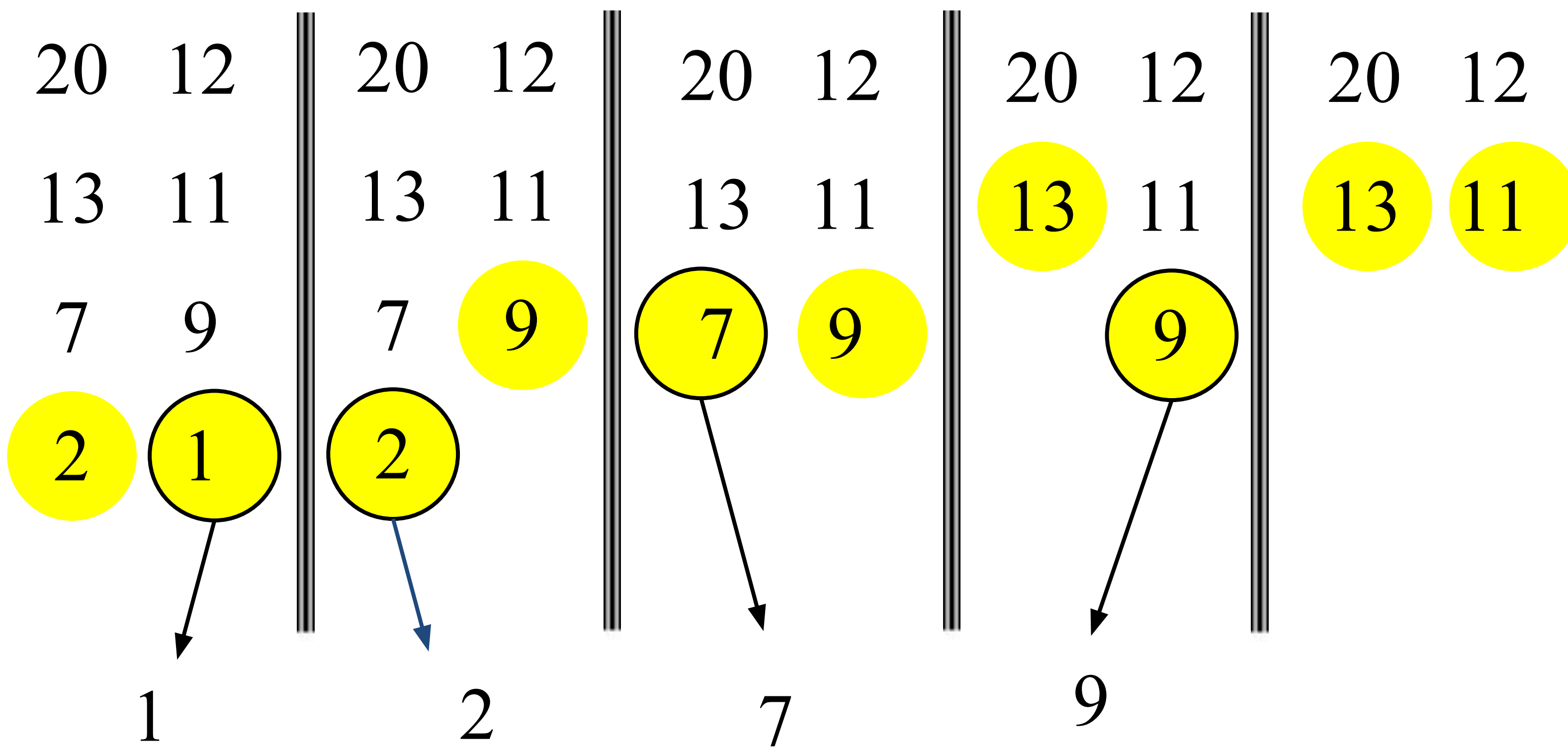
정렬된 두 배열 합병



정렬된 두 배열 합병

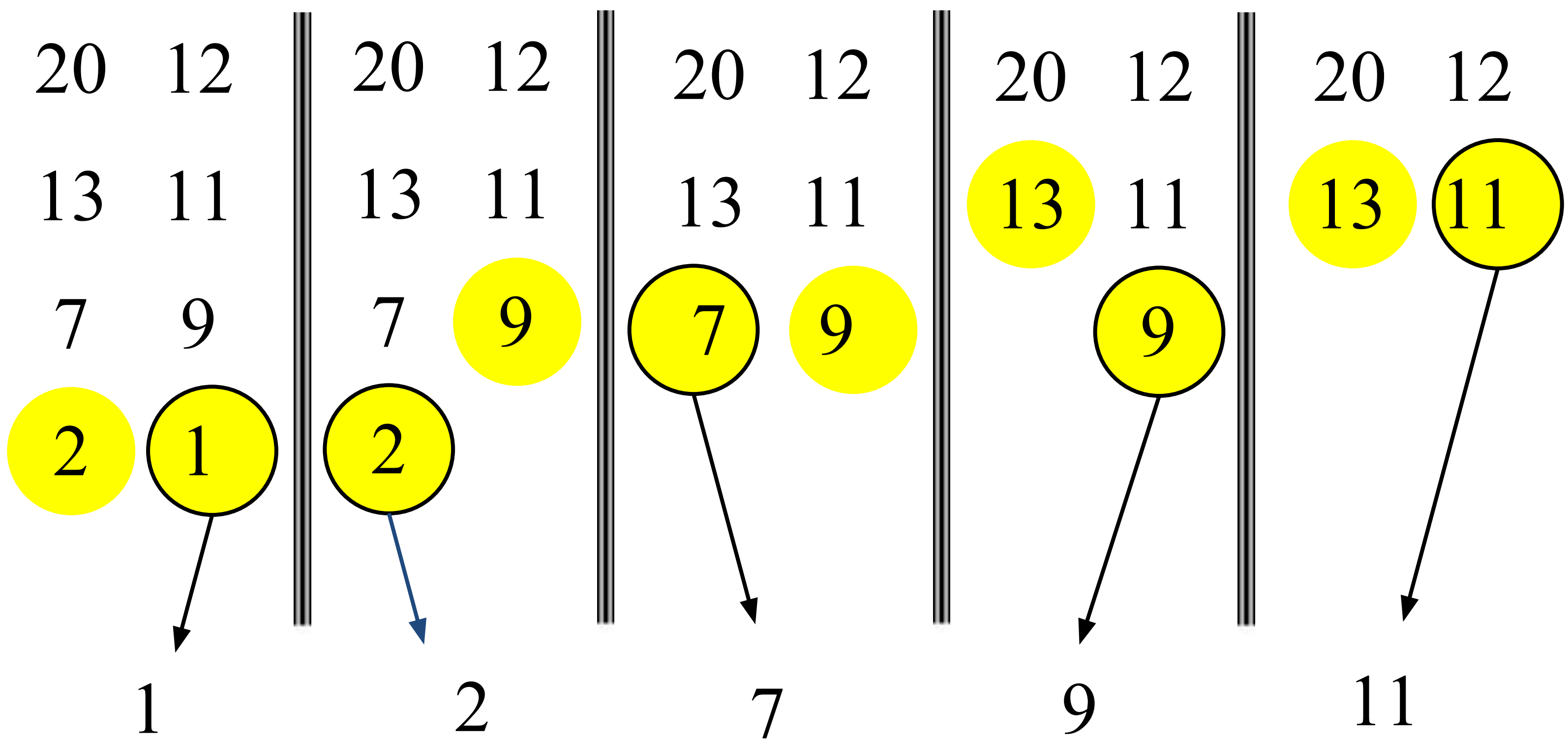


정렬된 두 배열 합병

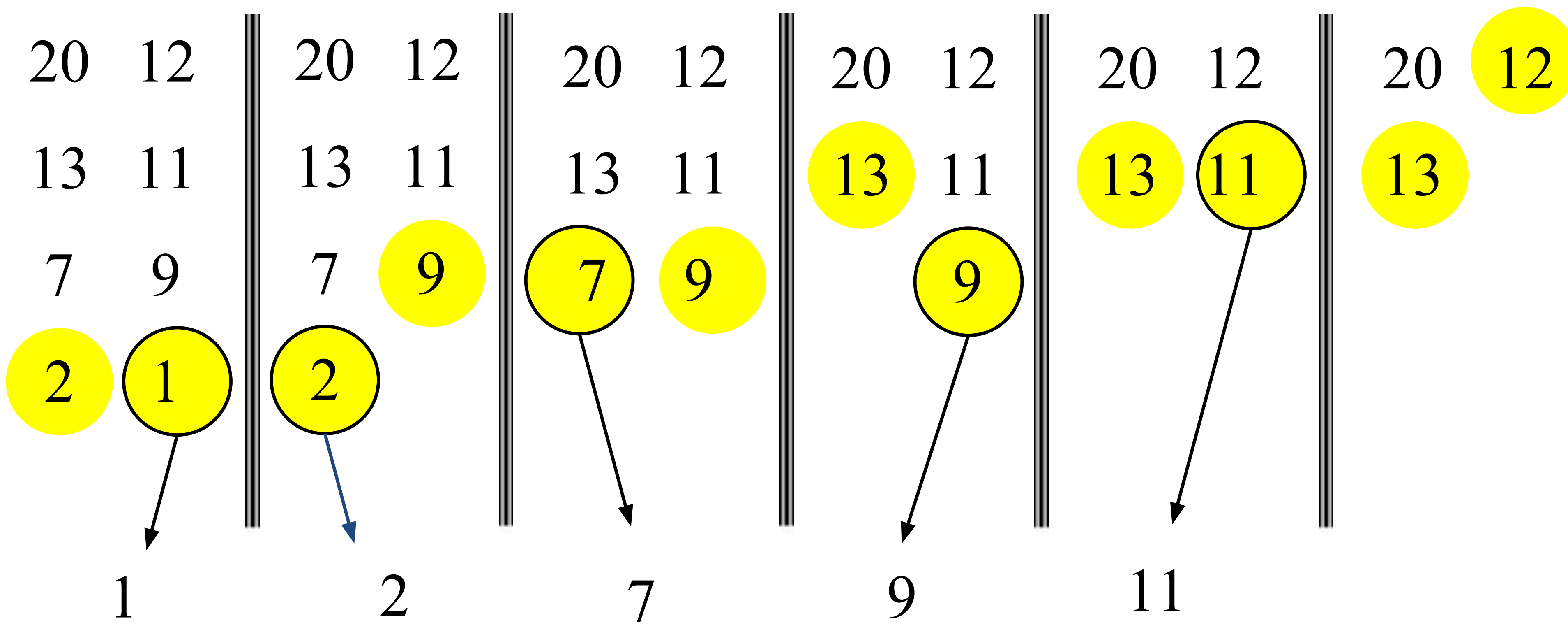


20 12
13 11
7 9
2 1
1
20 12
13 11
7 9
2
2
20 12
13 11
7 9
7
20 12
13 11
9
9
20 12
13 11

정렬된 두 배열 합병

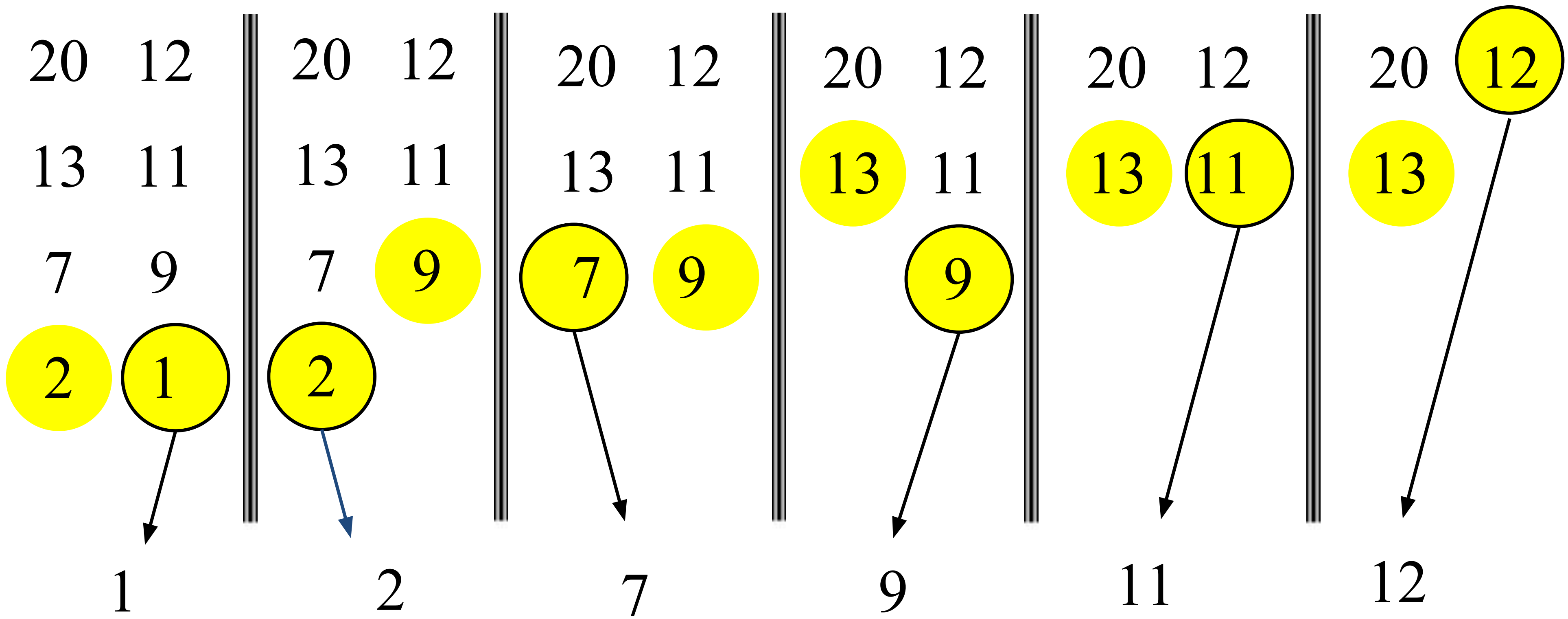

20 12
13 11
7 9
2 1
1
20 12
13 11
7 9
2
2
20 12
13 11
7 9
7
20 12
13 11
9
9
20 12
13 11
11

정렬된 두 배열 합병



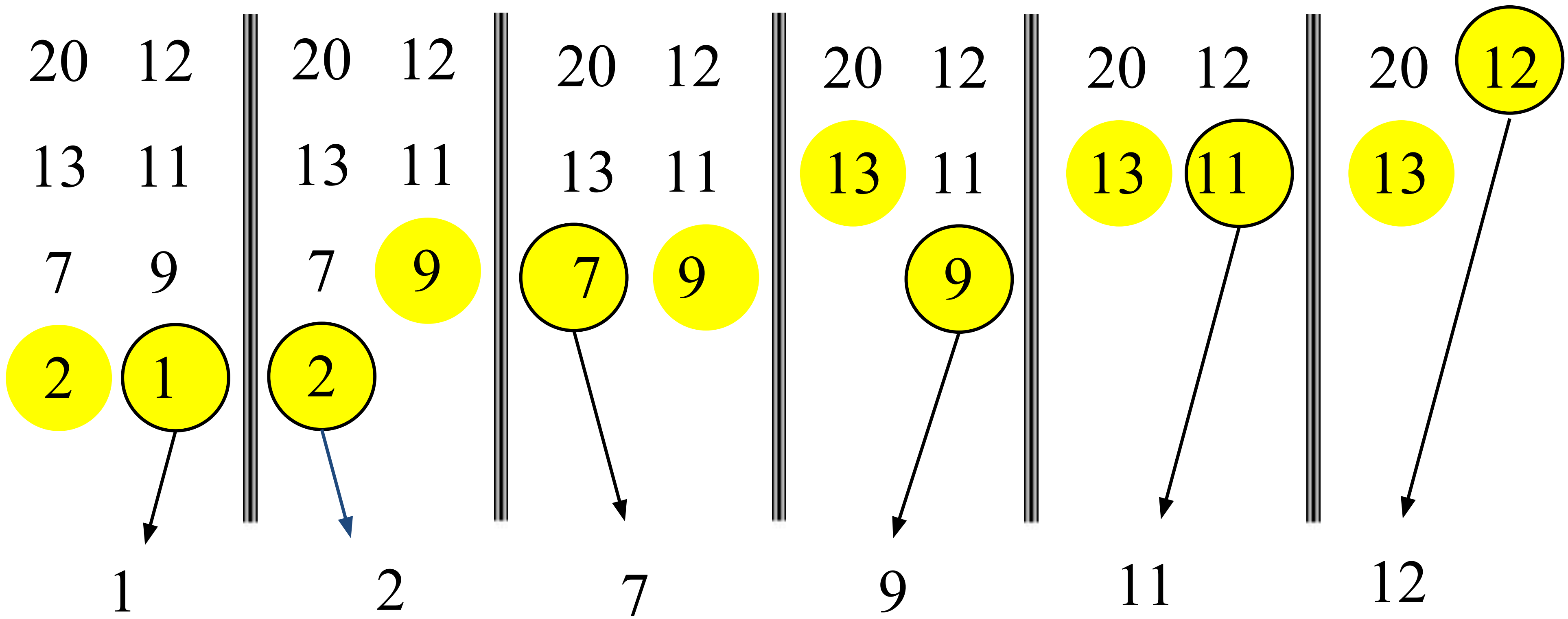
20 12
13 11
7 9
2 1
1
20 12
13 11
7 9
2
2
20 12
13 11
7 9
7
20 12
13 11
9
9
20 12
13 11
11
20 12
13

정렬된 두 배열 합병



20 12
13 11
7 9
2 1
1
20 12
13 11
7 9
2
2
20 12
13 11
7 9
7
20 12
13 11
9
9
20 12
13 11
11
20 12
13
12

정렬된 두 배열 합병



총 n 개의 항목을 합병하는 데
걸리는 시간 = $\Theta(n)$ (선형적 시간)

합병 정렬 분석

합병 정렬 $A[1\,..\,n]$ | $T(n)$

1. $n = 1$일 때, 바로 끝. | $\Theta(1)$
2. $A[\,1\,..\,\lceil n/2\rceil\,]$과 $A[\,\lceil n/2\rceil+1\,..\,n\,]$ 재귀적으로 합병. | $2T(n/2)$
3. 정렬된 리스트 두 개 **"합병"** | $\Theta(n)$

$$T(n) = \begin{cases} \Theta(1) \ (n = 1\text{일 때}); \\ 2T(n/2) + \Theta(n) \ (n > 1\text{일 때}); \end{cases}$$

$$T(n) = ?$$

재귀 풀이

$+c_1$ (divide)

풀이: $T(n) = \underbrace{2T(n/2)}_{\text{재귀}} + \underbrace{cn}_{\text{merge}}$ ($c > 0$은 상수)

재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)

$$T(n)$$

재귀 트리

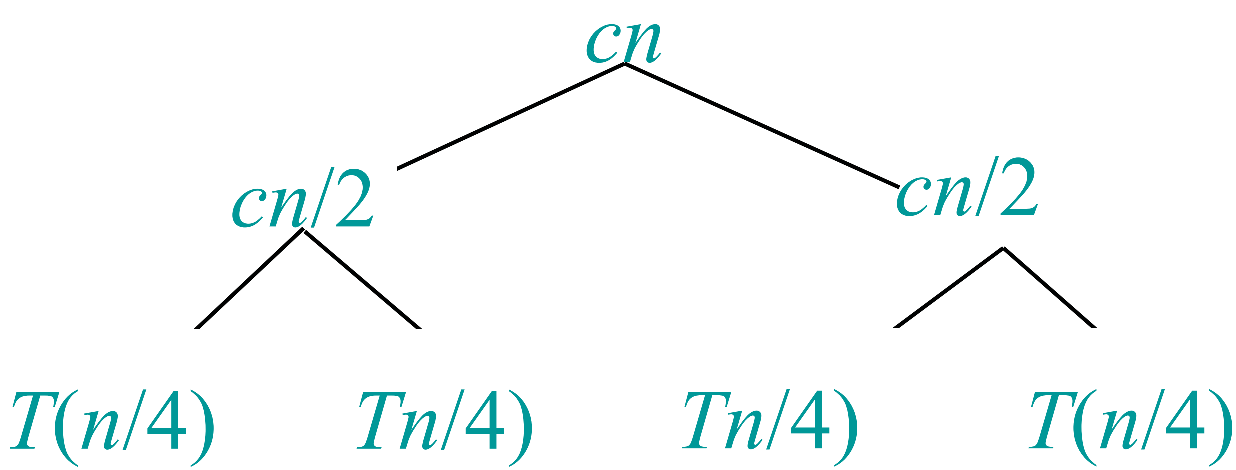
풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)

cn

$T(n/2)$ $\quad\quad$ $Tn/2)$

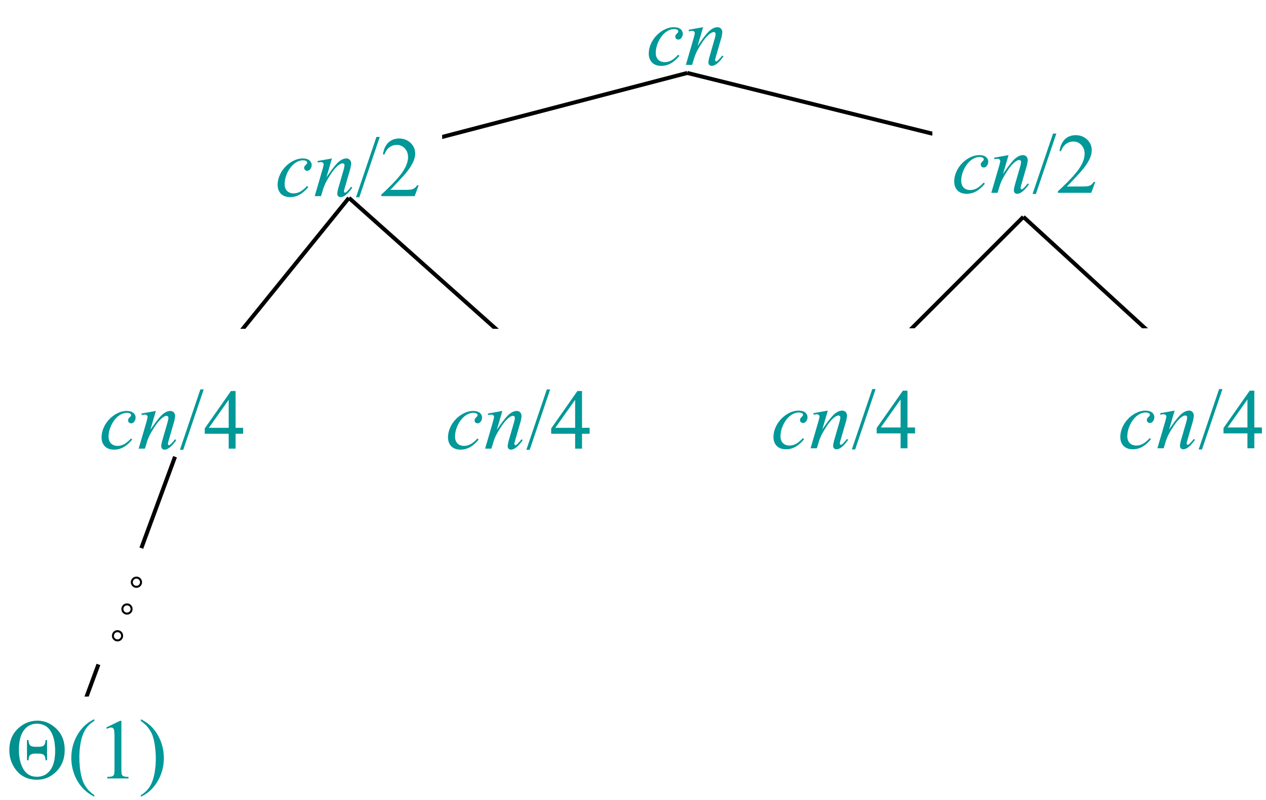
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



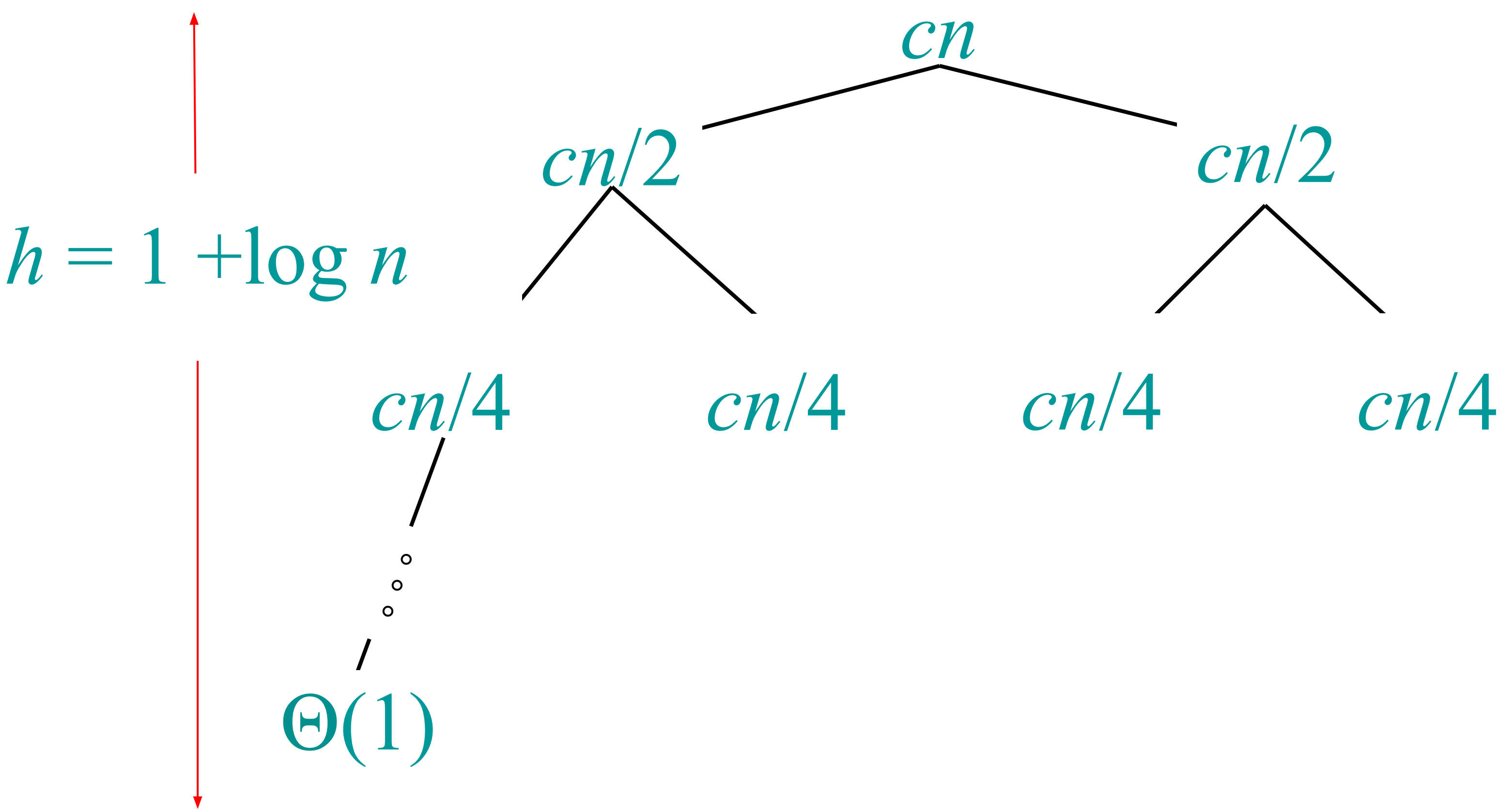
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



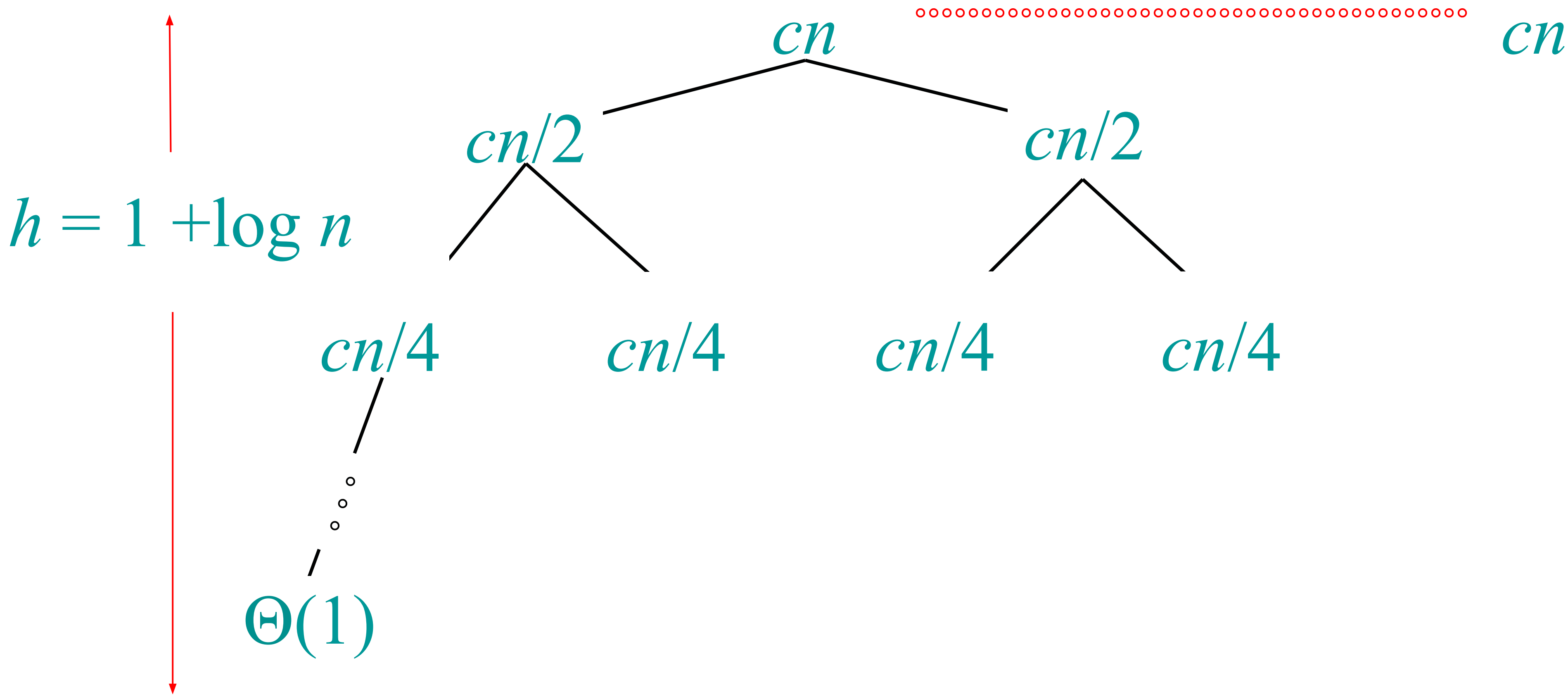
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



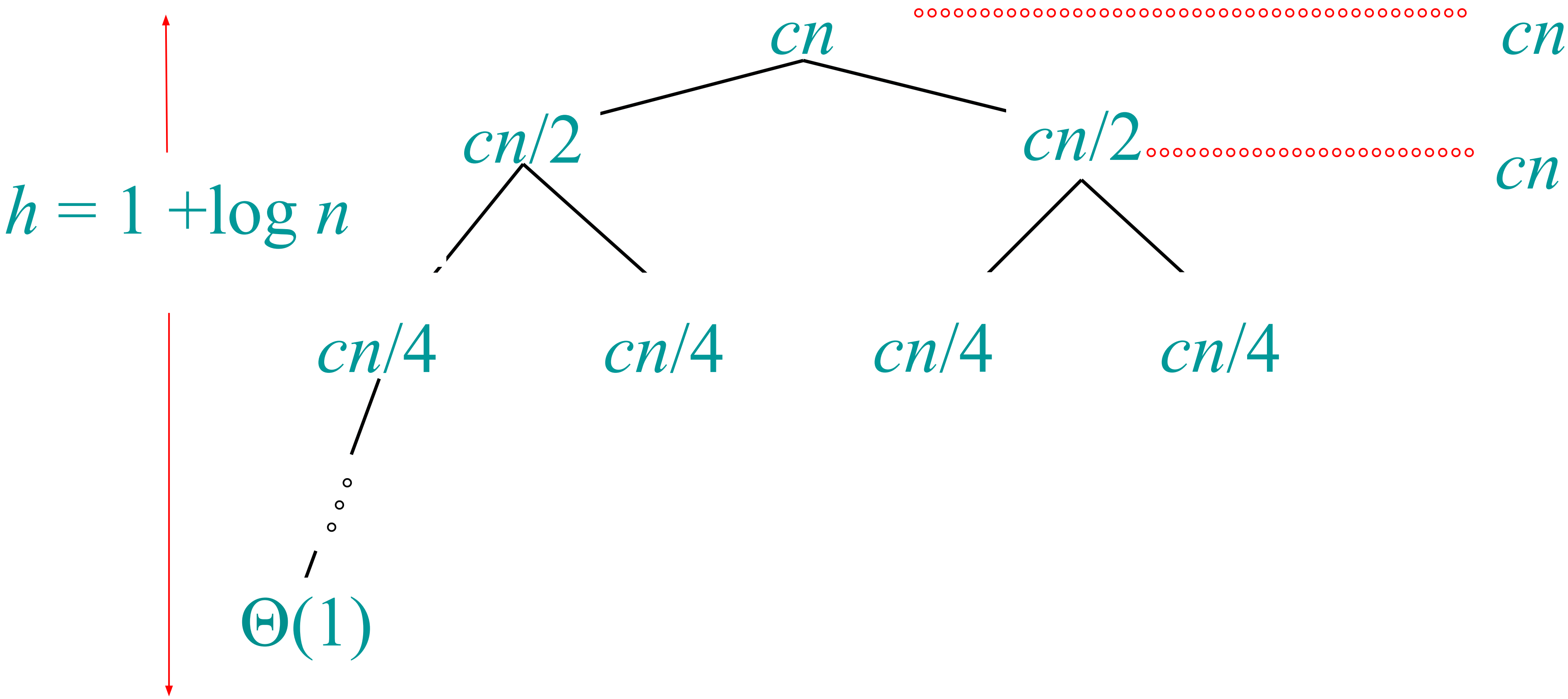
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



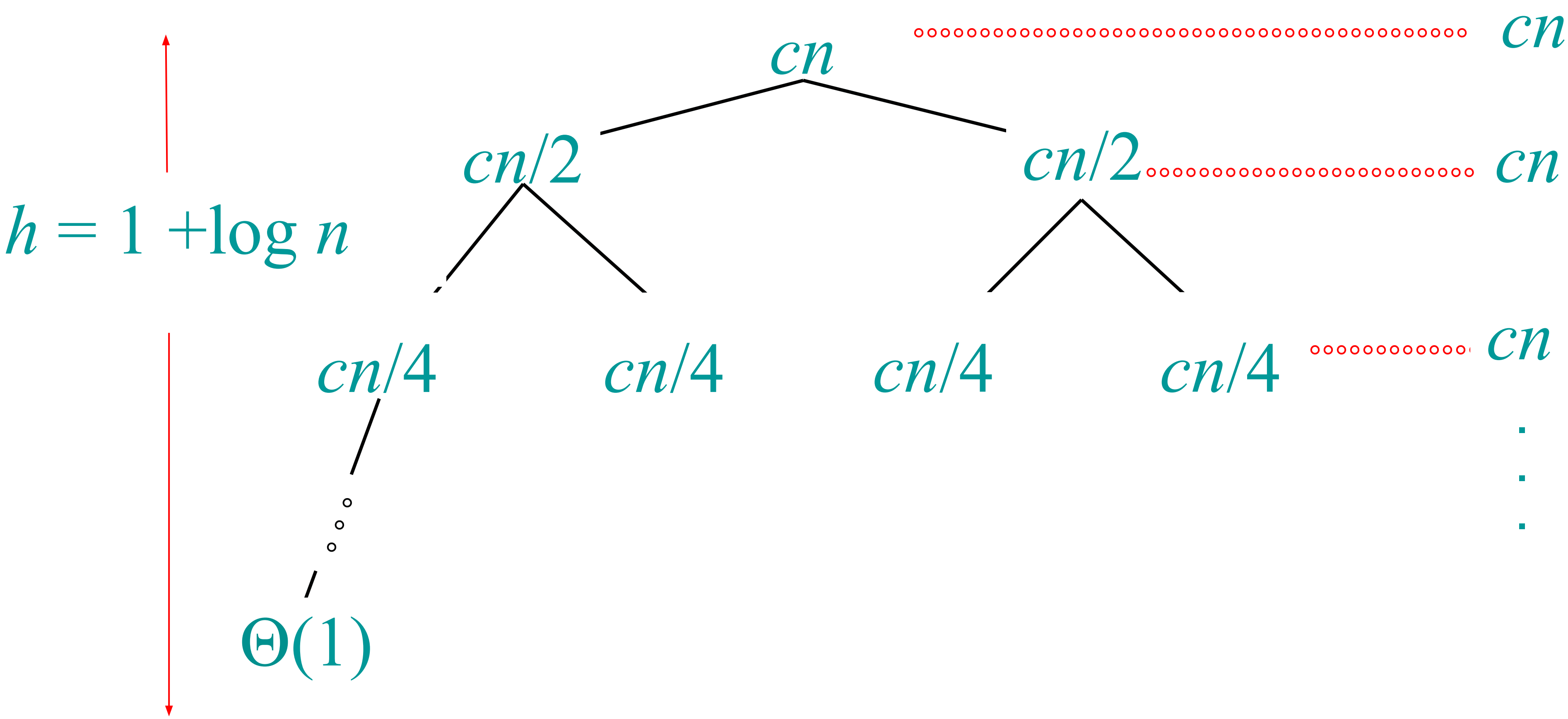
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



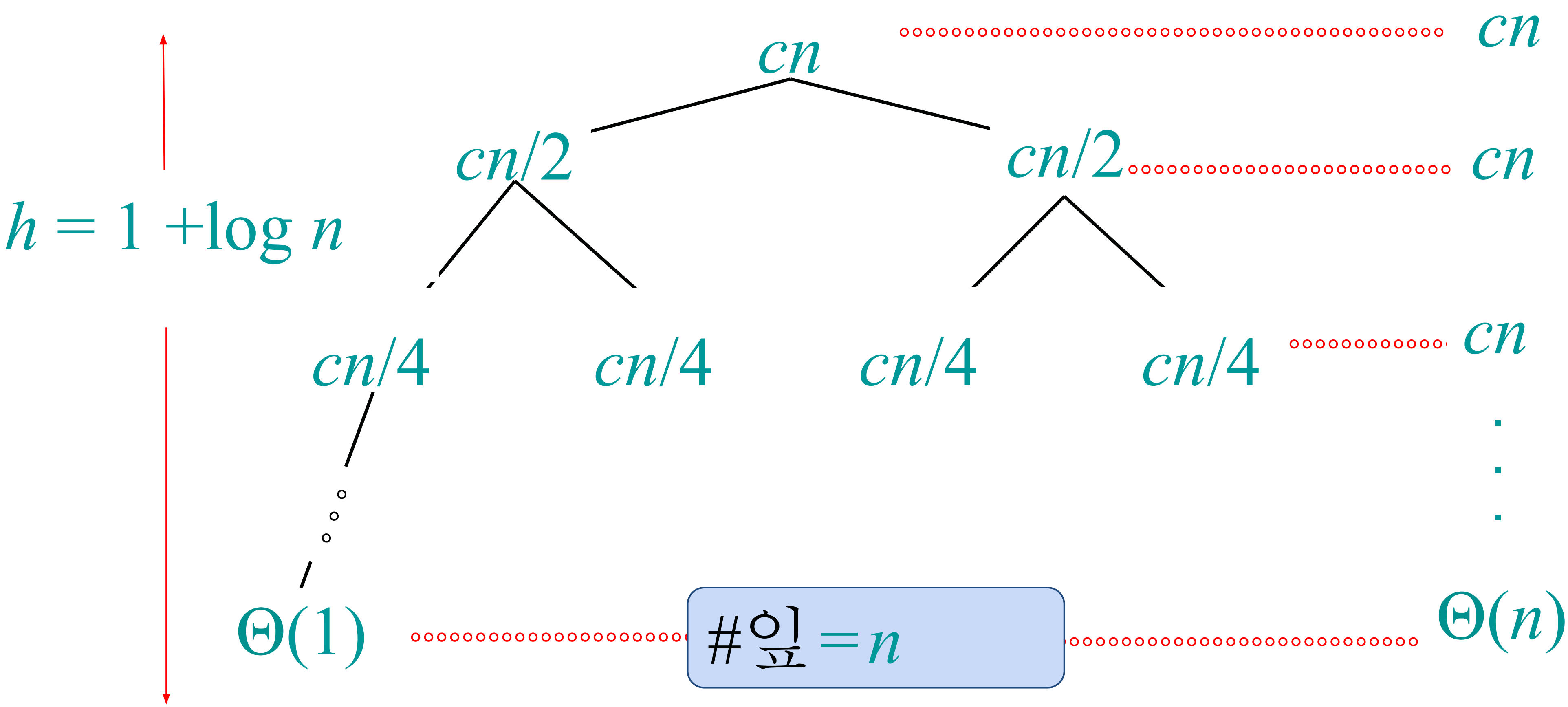
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



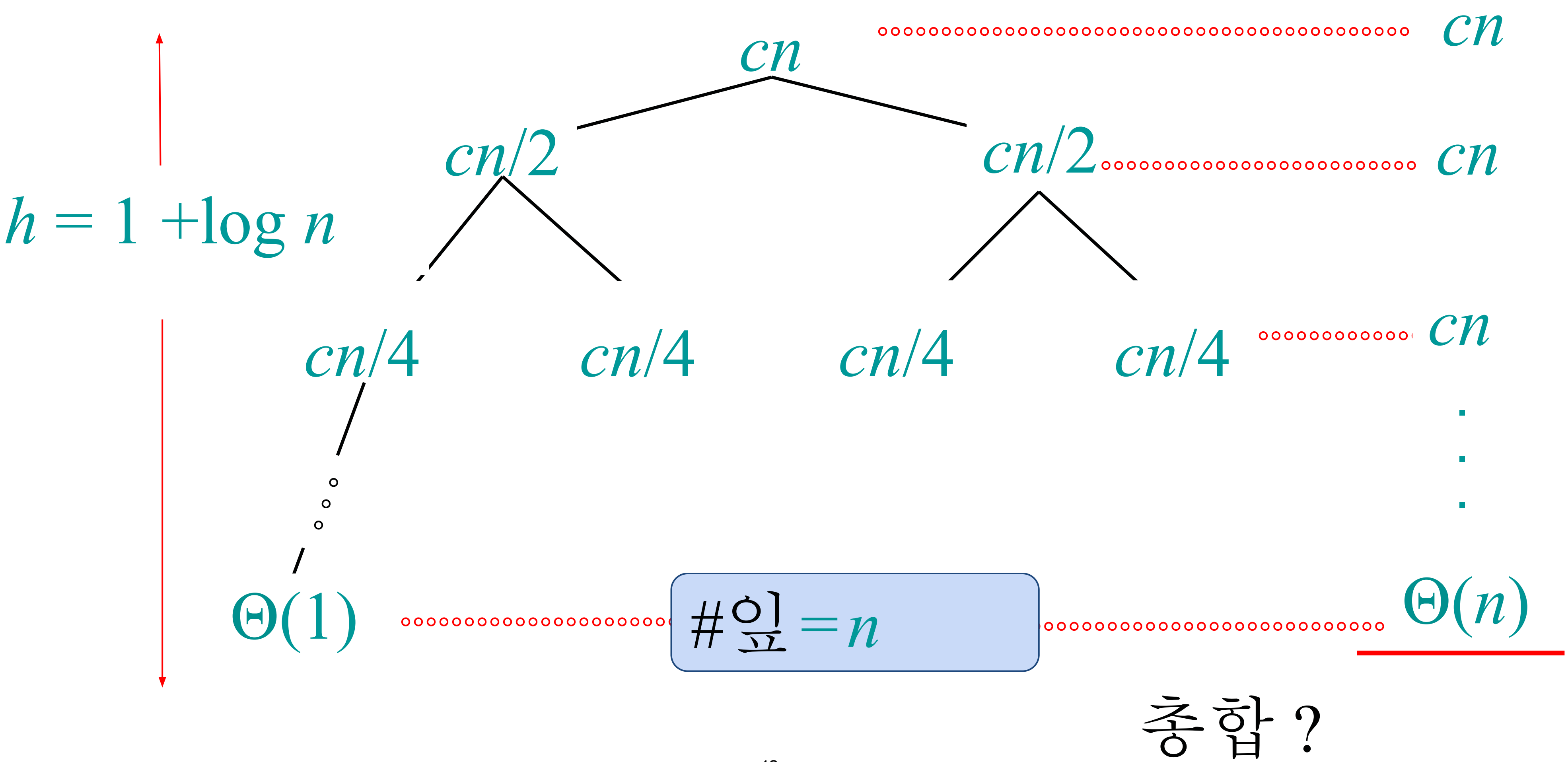
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



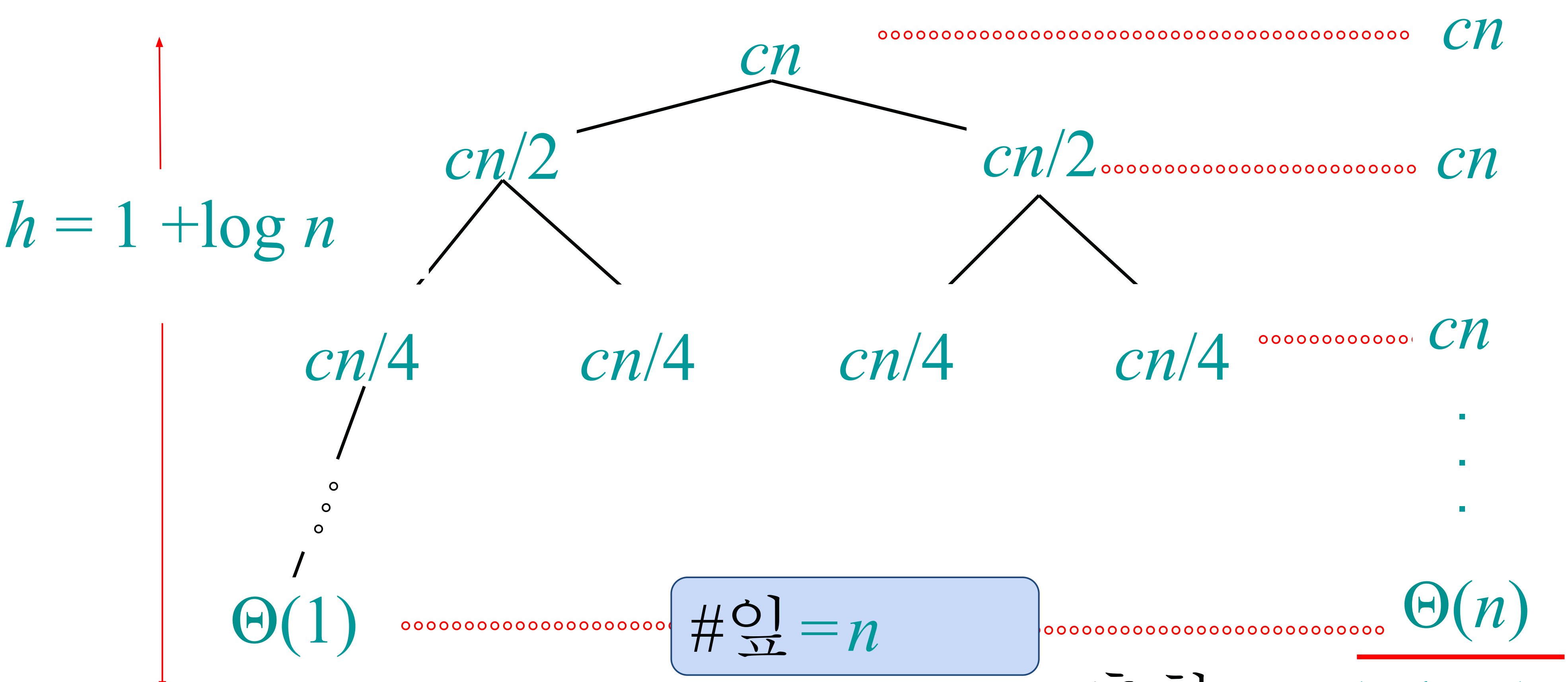
재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)



재귀 트리

풀이: $T(n) = 2T(n/2) + cn$ ($c > 0$은 상수)

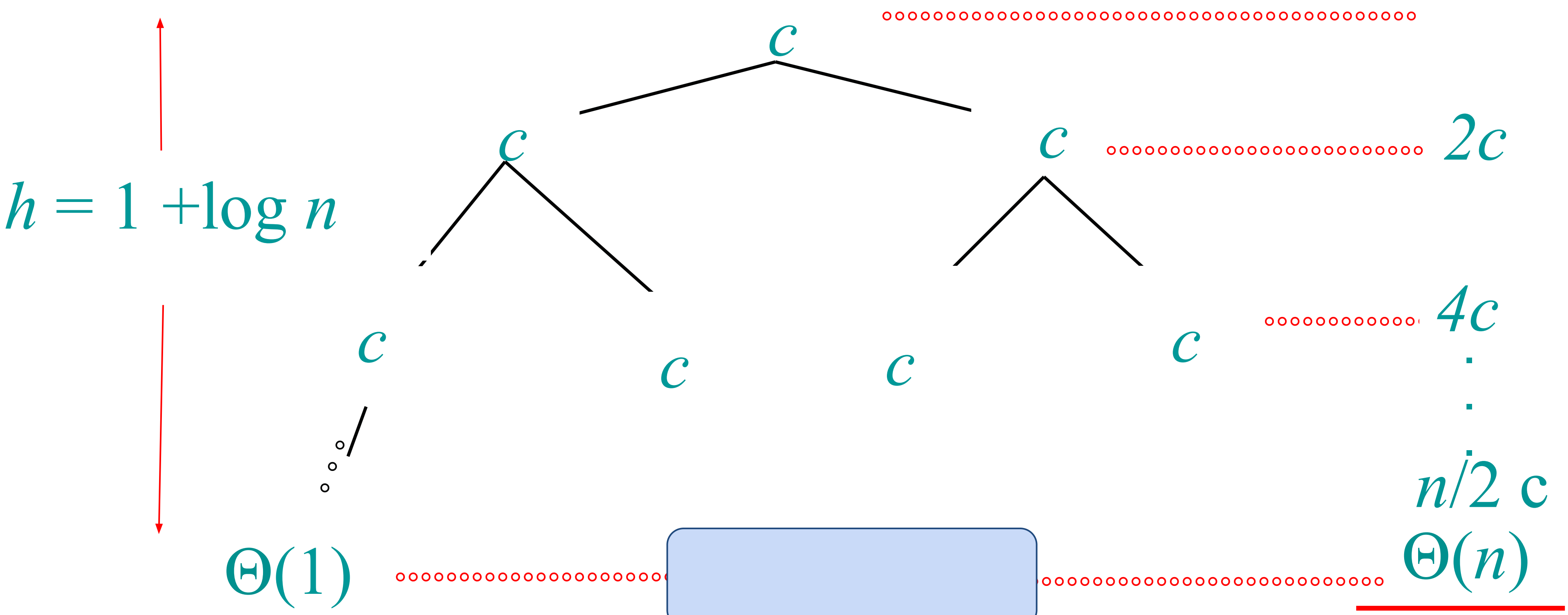


각 단계에서 작업의 양은 같다

총합 $= \Theta(n \lg n)$

다른 재귀에 대한 트리

풀이: $T(n) = 2T(n/2) + c$ ($c > 0$은 상수)



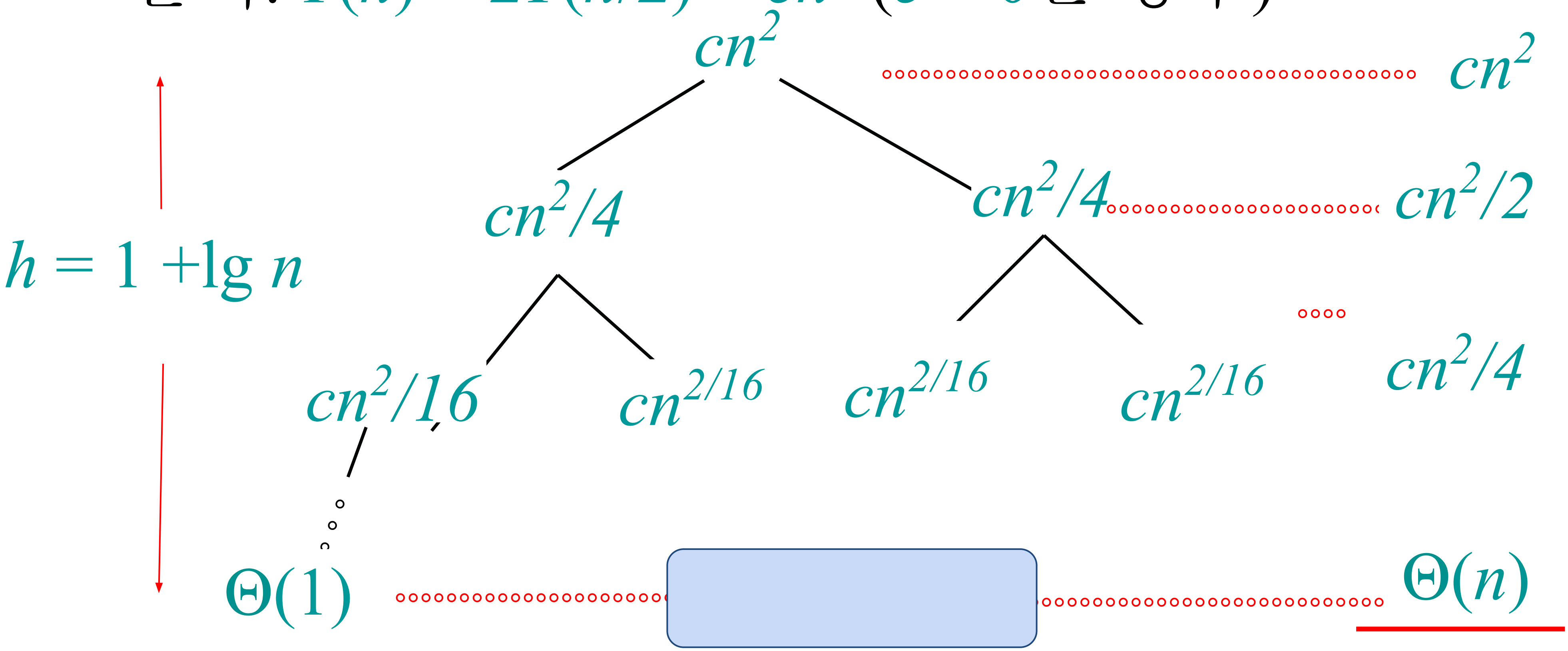
주의: $1 + \frac{1}{2} + \frac{1}{4} + \ldots < 2$

총합 $= \Theta(n)$

잎에서 일어나는 모든 작업

또 다른 재귀에 대한 트리

풀이: $T(n) = 2T(n/2) + cn^2$ ($c > 0$은 상수)



주의: $1 + ½ + ¼ + \ldots < 2$

총합 $= \Theta(n^2)$

루트에서 일어나는 모든 작업

MIT OpenCourseWare
http://ocw.mit.edu

6.006 알고리즘의 기초
가을 2011

$N \times \left(\frac{1}{2}\right)^h = 1$

$N = 2^h$

$\log N = h$

☆ 삽입정렬의 장점

= 추가 공간이 필요 없다.