제 7장 Recurrence Relations

7.1 Introduction

7.2 Solving Recurrence Relations

7.3 Applications to the Analysis of Algorithms

7.4 ~~The Closest-Pair Problems~~

7.1 Introduction

- 점화 관계는 수열을 선행 항으로 표현

- 정의 7.1.1 수열 $a_0, a_1, \ldots$ 에 대한 **점화 관계**는 a_n과 그 앞의 항들인 특정 $a_0, a_1, \ldots, a_{n-1}$과의 관계를 나타내는 식이다.
  수열 $a_0, a_1, \ldots$ 에 대한 **초기 조건**은 유한 개의 수열 항에 명시적으로 주어진 값들이다.

- 예제 7.1.2 피보나치 수열은 점화 관계

$$f_n = f_{n-1} + f_{n-2}, \qquad n \geq 3$$

  에 의해서 정의되며 , 초기 조건은

$$f_1 = 1, f_2 = 1$$

7.1 Introduction

- 예제 7.1.3 어떤 사람이 1000달러를 매년 12%의 복리로 투자한다. A_n이 n년 후의 연말 총액을 나타낸다고 할 때, 수열 $\{A_n\}$ 을 정의하는 점화 관계와 초기 조건을 구하라.

- 풀이 $n-1$년 후의 연말 총액은 A_{n-1}이다. 1년 후에는 총액 A_{n-1}에 이자를 더한 만큼 얻을 수 있다. 그러면

$$A_n = A_{n-1} + (0.12)A_{n-1} = (1.12)A_{n-1}, \qquad n \geq 1$$

이 점화관계를 $n = 1$에 적용하기 위해서는 A_0의 값을 알아야 한다.
A_0는 시작 총액이기 때문에 초기 조건은

$$A_0 = 1000.$$

임의의 값 n에 대해

$$A_n = (1.12)A_{n-1} = \cdots = (1.12)^n A_0 = (1.12)^n 1000$$

7.1 Introduction

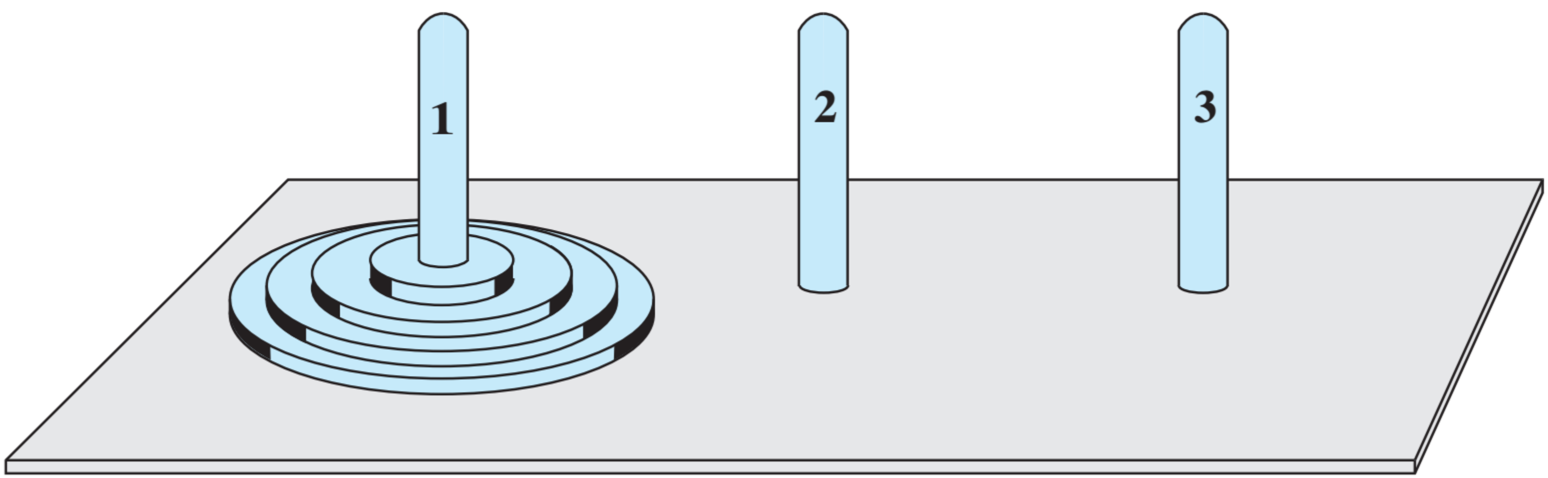
- 점화 관계, 재귀적 알고리즘, 수학적 귀납법은 밀접한 관련이 있다.

- Algorithm 7.1.4 복리계산
  이 재귀적 알고리즘은 초기값 1000달러에 매년 12%의 복리일 경우 n년 후의 연말 총액을 계산한다.
  Input: n, 연수
  Output: n년 후의 연말총액

1. $compound\_interest(n)$ {
2.   if (n == 0)
3.   return 1000
4.   return $1.12 * compound\_interest(n-1)$
5. }

7.1 Introduction

- 예제 7.1.8 하노이 탑은 판자에 세워진 3개의 막대와 중심에 구멍이 뚫린 다양한 크기의 원판 n개로 구성된 퍼즐
- 원판을 막대에 끼울 때, 기존 원판이 있을 경우 맨 위의 원판보다 지름이 작은 원판만을 끼울 수 있다고 한다.
- 그림과 같이 한 개의 막대에 주어진 모든 원판이 쌓여 있을 때, 원판을 한 장씩 움직여서 다른 막대로 원판을 모두 옮기는 문제



7.1 Introduction

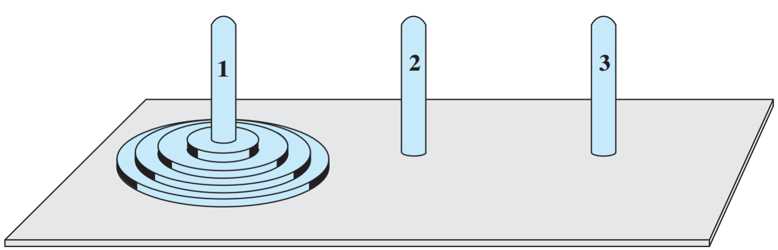
- 원판이 한 장만 있는 경우, 원하는 막대에 옮긴다.
- 막대 1에 $n > 1$개의 원판이 있는 경우

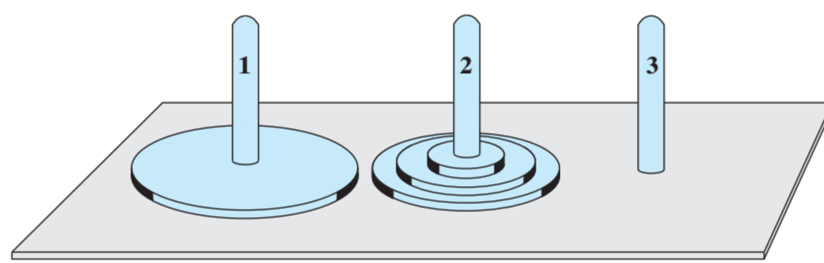
① 알고리즘을 재귀 호출하여 상위 $n{-}1$ 개의 원판을 막대 2로 움직인다. 움직이는 동안 막대 1의 제일 밑에 있는 원판은 그대로 남아있다.

② 막대 1에 남아 있는 원판을 막대 3으로 옮긴다.

③ 알고리즘을 재귀 호출하여 막대 2의 $n-1$ 개 원판을 막대 3 으로 옮긴다.

- c_n은 n개의 원판 퍼즐을 해결하는 데 필요한 원판 이동 횟수를 나타낸다
- $n > 1$인 경우, $(n-1)$개의 원판 퍼즐을 두 번 풀고, 명시적으로 원판 하나를 움직인다. 따라서 $c_n = 2c_{n-1} + 1, \ \ n > 1$, 초기 조건은 $c_1 = 1$.





7.2 Solving Recurrence Relations

- 일반 항 a_n에 대한 명시적 공식을 찾는 방법
  - Iteration(반복법)
  - Liner homogeneous recurrence relation with constant coefficients(상계수 선형 동차 점화관계)
- 반복법으로 수열 $a_0, a_1, \ldots$ 에 대한 점화 관계를 풀 때에는, a_n을 선행항 $a_{n-1}, \ldots, a_0$ 몇 개로 표현하는 데 점화 관계를 이용한다. 점화 관계를 순차적으로 사용하여 각각의 $a_{n-1}, \ldots$을 해당하는 선행 항 몇 개로 치환한다. 이러한 과정을 명시적 공식이 얻어질 때까지 계속한다.
- 예제 7.2.1 반복법으로 초기 조건은 $a_1 = 2$ 인 점화 관계 $a_n = a_{n-1} + 3$ 를 풀어라
- 풀이 $a_n = a_{n-1} + 3 = (a_{n-2} + 3) + 3 = a_{n-2} + 2 \cdot 3 = (a_{n-3} + 3) + 2 \cdot 3$

  $= a_{n-3} + 3 \cdot 3$

  일반적으로, $a_n = a_{n-k} + k \cdot 3$

  $k = n - 1$라 두면, $a_n = a_1 + (n-1) \cdot 3 = 2 + 3(n-1)$

7.2 Solving Recurrence Relations

- 예제 7.2.2 점화 관계 $S_n = 2S_{n-1}$, 초기 조건 $S_0 = 1$ 반복법으로 풀어라
- 풀이 $S_n = 2S_{n-1} = 2(2S_{n-2}) = \cdots = 2^n S_0 = 2^n$.
- 예제 7.2.3 개체수 증가

  어느 지역의 사슴 수를 시각 $n = 0$일 때 1000이라 가정하자. 그리고 시각이 $n-1$에서 n으로 바뀔 때 그 수는 $n-1$ 시각 개체수의 10% 만큼 증가한다. 점화 관계와 초기 조건을 구하고, 점화 관계를 풀라.

- 풀이 d_n을 시각이 n일 때 사슴의 수라 하자.

  초기 조건은 $d_0 = 1000$

  시각 $n-1$ 에서 n까지의 개체수 증가는 $d_n - d_{n-1}$이다. 이 증가는 시각이 $n-1$일 때 크기의 10% 이므로 다음 점화 관계를 얻는다

$$d_n - d_{n-1} = 0.1d_{n-1} \quad \text{or} \quad d_n = 1.1d_{n-1}$$

  점화 관계는 다음과 같이 반복법으로 풀 수 있다.

$$d_n = 1.1d_{n-1} = 1.1(1.1d_{n-2}) = (1.1)^2 d_{n-2} = \cdots = (1.1)^n d_0 = (1.1)^n\, 1000$$

7.2 Solving Recurrence Relations

- 예제 7.2.4 원판 n개의 하노이 Tower에서, 원판 이동 최소 회수 c_n 에 대해 명시적 공식을 구하라. 점화 관계 $c_n = 2c_{n-1} + 1$, 초기 조건 $c_1 = 1$.
- 풀이 반복법을 적용하면,

$$
\begin{aligned}
c_n &= 2c_{n-1} + 1 \\
&= 2(2c_{n-2} + 1) + 1 \\
&= 2^2 c_{n-2} + 2 + 1 \\
&= 2^2(2c_{n-3} + 1) + 2 + 1 \\
&= 2^3 c_{n-3} + 2^2 + 2 + 1 \\
&\quad \cdots \\
&= 2^{n-1} c_1 + 2^{n-2} + 2^{n-3} + \cdots + 2^1 + 1 \\
&= 2^{n-1} + 2^{n-2} + 2^{n-3} + \cdots + 2^1 + 1 \\
&= 2^n - 1. \text{ (by 등비 수열의 합)}
\end{aligned}
$$

7.2 Solving Recurrence Relations

- 정의 7.2.6 Order k(k 차) 상계수 선형 동차 점화 관계는 다음과 같은 형태이다.

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}, \qquad c_k \neq 0 \qquad (2.5)$$

- k개 초기 조건

$$a_0 = C_0, \qquad a_1 = C_1, \qquad a_{k-1} = C_{k-1}$$

이 설정된 k차 상계수 선형 동차 점화 관계 (2.5)는 수열 $a_0, a_1, \ldots$ 을 유일하게 정의한다.

- 예제 7.2.7 예제 7.2.2의 점화 관계

$$S_n = 2S_{n-1}$$

는 1차 상계수 선형 동차 점화 관계이다.

- 피보나치 수열을 정의한 점화 관계

$$f_i = f_{i-1} + f_{i-2}$$

는 2차 상계수 선형 동차 점화 관계이다.

7.2 Solving Recurrence Relations

- 정리 7.2.11(a) 다음식을 2차 상계수 선형 동차 점화 관계라 하자.

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} \qquad (2.13)$$

만약 $\mathrm{p_n}$과 $\mathrm{q_n}$ 이 (2.13) 의 해라면,

$a_n = \mathrm{ap_n} + \mathrm{bq_n}$ where a, b 는 상수 (2.14) 역시 (2.13)의 해이다.

만일 r이

$$\mathrm{r}^2 - c_1\mathrm{r} - c_2 = 0, \qquad (2.15)$$

의 근이라면, $n = 0, 1, \ldots,$ 에 대한 수열 r^n은 식 (2.13) 의 기본해이다.

$a_0 = \mathrm{d_0},\ a_1 = \mathrm{d_1}$ 의 초기 조건이 2차 상계수 동치 점화관계 (2.13)과

같이 주어지고, (2.15)의 근이 $\mathrm{r_1}, \mathrm{r_2}$라면

$a_n = \mathrm{j}r_1^n + \mathrm{k}r_2^n$ 이 성립하는 j, k가 정해 진다.

7.2 Solving Recurrence Relations

- 증명 $a_n = p_n$과 $a_n = q_n$ 이 (2.13)의 해이므로
  $p_n = c_1 p_{n-1} + c_2 p_{n-2}$, $q_n = c_1 q_{n-1} + c_2 q_{n-2}$ 을 만족한다
  (2.14)로 부터
  $a_{n-1} = ap_{n-1} + bq_{n-1}$ 와 $a_{n-2} = ap_{n-2} + bq_{n-2}$ (2.15)를 만족한다.
  (2.13)의 우변에 (2.15)를 적용하면
  $c_1 a_{n-1} + c_2 a_{n-2} = c_1(ap_{n-1} + bq_{n-1}) + c_2(ap_{n-2} + bq_{n-2})$
  $= a(c_1 p_{n-1} + c_2 p_{n-2}) + b(c_1 q_{n-1} + c_2 q_{n-2}) = ap_n + bq_n = a_n$ 이 성립한다.

- 만일 $a_n = dr^n$ 이라면
  식(2.13)은 $dr^n = dc_1 r^{n-1} + dc_2 r^{n-2}$ 이 성립하므로

  $$r^2 - c_1 r - c_2 = 0, \qquad (2.16)$$

  특성 방정식 (2.16)을 만족하는 근 r_1과 r_2 는 2차 방정식의 해이다.
  따라서, $a_n = jr_1^n + kr_2^n$ 이 성립한다.
  이때, $a_0 = d_0$, $a_1 = d_1$ 의 초기 조건이 2차 상계수 동치 점화관계와 같이 주어진다면, j, k 를 결정할 수 있다.

7.2 Solving Recurrence Relations

- 예제 7.2.12 어떤 지역의 사슴의 수가 시각 $n = 0$ 일 때 200, $n = 1$ 일 때 220 이라 하고, 시각 $n - 1$ 에서 n까지의 증가는 $n - 2$에서 $n - 1$까지의 증가의 2배라고 가정하자. 시각 n 일 때의 사슴의 수를 정의하고 점화 관계를 풀어라.

- 풀이 d_n을 시각이 n일 때 사슴의 수라 하자
  초기 조건은 $d_0 = 200, \ \ d_1 = 220.$
  점화 관계는
  $$d_n - d_{n-1} = 2(d_{n-1} - d_{n-2}) \quad \text{or} \quad d_n = 3d_{n-1} - 2d_{n-2}$$
  $\mathrm{r}^2 - 3\mathrm{r} + 2 = 0$의 근은 1과 2.

  $d_n = \mathrm{j} \cdot 1^n + \mathrm{k} \cdot 2^n = \mathrm{j} + \mathrm{k}2^n.$

  $200 = d_0 = \mathrm{j} + \mathrm{k}, \ 220 = d_1 = \mathrm{j} + \mathrm{k}2. \quad \therefore \ \mathrm{j} = 180, \ \mathrm{k} = 20$

  따라서 $d_n = 180 + 20 \cdot 2^n.$
  개체수 증가는 지수 함수이다.

7.2 Solving Recurrence Relations

- 예제 7.2.13 피보나치 수열에 대한 명시적 공식을 구하라.

  $$f_n = f_{n-1} + f_{n-2},\ n \geq 3,\ \text{초기 조건 } f_1 = 1,\ f_2 = 1$$

- 풀이 $\mathrm{r}^2 - \mathrm{r} - 1 = 0.\ \therefore \mathrm{r} = \frac{1 \pm \sqrt{5}}{2}.$

$$f_n = \mathrm{j}\left(\frac{1+\sqrt{5}}{2}\right)^n + \mathrm{k}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

$$\begin{cases} f_1 = \mathrm{j}\left(\frac{1+\sqrt{5}}{2}\right) + \mathrm{k}\left(\frac{1-\sqrt{5}}{2}\right) = 1 \\ f_2 = \mathrm{j}\left(\frac{1+\sqrt{5}}{2}\right)^2 + \mathrm{k}\left(\frac{1-\sqrt{5}}{2}\right)^2 = 1 \end{cases} \quad \therefore \mathrm{j} = \frac{\mathbf{1}}{\sqrt{5}}, \mathrm{k} = -\frac{1}{\sqrt{5}}$$

$$f_n = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

7.2 Solving Recurrence Relations

- 정리 7.2.14 다음을 2차 상계수 선형 동차 점화 관계라 하자.

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} \qquad (2.16)$$

또한, 수열 a는 (2.16)과

$$a_0 = C_0, \qquad a_1 = C_1 \qquad (2.15)$$

을 만족시킨다고 하자. 만약

$$\mathrm{r}^2 - c_1 \mathrm{r} - c_2 = 0 \qquad (2.17)$$

의 두 근이 r로 같다면 $n = 0, 1, \ldots$ 대해서 $a_n = \mathrm{j}r^n + \mathrm{k}nr^n$ 인 상수 j과 k이 존재한다.

- 증명 수열 $nr^n, n = 0, 1, \ldots,$ 이 (2.16)의 해임을 보이겠다.

r이 (2.17)의 중근이므로

$$t^2 - c_1 t - c_2 = (t-r)^2 = t^2 - 2rt + r^2.$$

그러므로 $c_1 = 2r,\ c_2 = -r^2.$

$$c_1[(n-1)r^{n-1}] + c_2[(n-2)r^{n-2}] = 2r[(n-1)r^{n-1}] - r^2[(n-2)r^{n-2}]$$

$$= r^n\,[2(n-1) - (n-2)] = nr^n.$$

7.2 Solving Recurrence Relations

- 예제 7.2.15 초기 조건이 $d_0 = 1 = d_1$, 점화 관계는

$$d_n = 4(d_{n-1} - d_{n-2}) \qquad (2.18)$$

- 풀이 정리 7.2.11에 따라 $S_n = r^n$ 는 (2.18)의 해이다. 여기서 r은

$$r^2 - 4r + 4 = 0 \qquad (2.19)$$

의 해이다. 따라서 (2.18)의 해는 $S_n = 2^n$이다. 2는 (2.19)의 유일해(중근)이므로 정리 7.2.14에 의해

$$d_n = n2^n$$

도 (2.18)의 해이다. 그러므로 (2.18) 대한 일반해 형식은

$$d_n = j2^n + kn2^n$$

이다. 여기서

$$d_0 = 1 = d_1$$

이어야 한다. 이 마지막 등식들은

$$d_0 = j + k0 = 1, \qquad d_1 = j2 + k2 = 1.$$

이 된다. j, k에 대해 풀면

$$j = 1, \qquad k = -\frac{1}{2}$$

그러므로 (2.18)의 해는 $d_n = 2^n - n2^{n-1}$

7.3 Applications to the Analysis of Algorithms

- 선택 정렬 알고리즘은 다음 수열을 비내림차순으로 정렬하는데

$$s_1, s_2, \dots, s_n$$

  먼저 최대 항목을 선택하여 이 것을 맨 뒤로 옮긴 다음, 나머지 항목을 재귀적으로 정렬한다.

- b_n는 n개 항목 정렬에 필요한 알고리즘 8번 행에서 비교 실행 횟수이다

$$b_1 = 0, \qquad b_n = (n-1) + b_{n-1}$$

  반복법,

$$
\begin{aligned}
b_n &= b_{n-1} + n - 1 \\
&= (b_{n-2} + n - 2) + (n-1) = b_{n-2} + (n-2) + (n-1) \\
&= (b_{n-3} + n - 3) + (n-2) + (n-1) = b_{n-3} + (n-3) + (n-2) + (n-1) \\
&\ \vdots \\
&= b_1 + 1 + 2 + \cdots + (n-2) + (n-1) \\
&= 0 + 1 + 2 + \cdots + (n-2) + (n-1) = \frac{(n-1)n}{2} = \Theta(n^2)
\end{aligned}
$$

7.3 Applications to the Analysis of Algorithms

- 알고리즘 7.3.1 선택정렬
- Input: 길이 n 수열 $s_1, s_2, \dots, s_n$

  Output: 비감소(nondecreasing)순으로 정렬된 길이 n 수열 $s_1, s_2, \dots, s_n$

$selection\_sort(s, n)$ {
  if ($n == 1$)
    return
  //최대값을 찾는다
  $max\_index = 1$ //처음에는 s_1을 최대값으로 가정 한다
  for $i = 2$ to n
    if ($s_i > s_{max\_index}$) //큰 값이 있으면 수정 한다
      $max\_index = i$
  //최대값을 마지막으로 옮긴다
  $swap(s_n, s_{max\_index})$
  $selection\_sort(s, n - 1)$
}

$$\boxed{\begin{aligned} b_1 &= 0, \\ b_n &= (n-1) + b_{n-1} \end{aligned}}$$

7.3 Applications to the Analysis of Algorithms

- 이진 탐색은 정렬된 수열에서 값을 찾고, 값이 발견되면 값의 색인을 반환하며 값이 발견되지 않으면 0을 돌려준다. 이 알고리즘은 분할정복(divide-and-conquer)법을 사용한다.

- 이진 탐색에서 최악의 경우 소요 시간을 n항 수열에 대해서 최악의 경우로 알고리즘이 호출되는 횟수로 정의하고, a_n으로 표현

- $a_1 = 2$
  $n > 1$일 때, 원래 수열의 왼쪽의 크기는 $\lfloor (n-1)/2 \rfloor$ 이고 오른쪽은 $\lfloor n/2 \rfloor$ 이며, 최악의 경우는 더 긴 수열에서 나타나기 때문에 총 호출 횟수는 $a_{\lfloor n/2 \rfloor}$ 이 된다.

$$a_n = 1 + a_{\lfloor n/2 \rfloor}$$

7.3 Applications to the Analysis of Algorithms

- 알고리즘 7.3.2 이진탐색

- Input: 비감소순으로 정렬된 수열 $s_i, s_{i+1}, \ldots, s_j$, $i \geq 1$, 찾을 값 key, i, j
  Output: 출력은 $s_k = key$ 이면 색인 k 이고, key가 수열에 없으면 0 이다.

$binary\_search(s, i, j, key)$ {
  if $(i > j)$ // 찾을 수 없다
    return 0
  $k = \lfloor (i + j)/2 \rfloor$
  if $(key == s_k)$ // 찾았다
    return k
  if $(key < s_k)$ // 왼쪽 반에서 찾는다
    $j = k - 1$
  else // 오른쪽 반에서 찾는다
    $i = k + 1$
  return $binary\_search(s, i, j, key)$
}

$$\boxed{\begin{aligned} a_1 &= 2 \\ a_n &= 1 + a_{\lfloor n/2 \rfloor} \end{aligned}}$$

7.3 Applications to the Analysis of Algorithms

- $a_1 = 2, a_n = 1 + a_{\lfloor n/2 \rfloor}$

  만약 $n = 2^k$이면,

$$\begin{aligned} a_n = a_{2^k} &= 1 + a_{2^{k-1}} \\ &= 1 + \left(1 + a_{2^{k-2}}\right) = 2 + a_{2^{k-2}} \\ &\cdots \\ &= k + a_{2^0} = k + 2 = \lg n + 2 \end{aligned}$$

  임의의 n은 두 개의 2의 거듭제곱 사이에 존재한다

$$2^{k-1} < n \le 2^k$$

  수열 a는 비감소 수열이므로

$$a_n \le a_{2^k} = \lg 2^k + 2 = \lg 2^{k-1} + 3 < \lg n + 3 \ \therefore\ a_n = O(\lg n)$$

$$a_n \ge a_{2^{k-1}} = \lg 2^{k-1} + 2 = \lg 2^k + 1 > \lg n \ \therefore\ a_n = \Omega(\lg n)$$

$\therefore$ $a_n = \Theta(\lg n)$

- 정리 7.3.4 입력 크기 n인 이진 탐색에서 최악의 경우 소요시간은 $\Theta(\lg n)$