

문제 01

축소 정복을 가장 직접적으로 나타내는 구조는?

- 1 항상 두 부분 문제를 모두 푼다
- 2 모든 중간 결과를 반드시 표에 저장한다
- 3 가능한 모든 완성 답을 생성한다
- 4 현재 가장 이익이 큰 선택만 확정한다
- 5 더 작은 한 부분 문제를 풀고 원래 답을 구성한다

문제 01 정답과 해설

축소 정복을 가장 직접적으로 나타내는 구조는?

5번 더 작은 한 부분 문제를 풀고 원래 답을 구성한다

해설

핵심은 같은 유형의 더 작은 한 문제로 이어지는 구조다. 재귀나 표의 사용 여부만으로 분류하지 않는다.

문제 02

$T(n)=T(n-1)+\Theta(1)$, $T(0)=\Theta(1)$ 일 때 가장 정확한 점근적 차수는?

1 $\Theta(1)$

2 $\Theta(n^2)$

3 $\Theta(n)$

4 $\Theta(n \log n)$

5 $\Theta(\log n)$

문제 02 정답과 해설

$T(n)=T(n-1)+\Theta(1)$, $T(0)=\Theta(1)$ 일 때 가장 정확한 점근적 차수는?

3번 $\Theta(n)$

해설

상수 작업이 n 단계 누적되므로 선형이다.

문제 03

$T(n)=T(n/2)+\Theta(n)$, $T(1)=\Theta(1)$, $n=2^k$ 일 때 차수는?

1 $\Theta(2^n)$

2 $\Theta(n^2)$

3 $\Theta(\log n)$

4 $\Theta(n)$

5 $\Theta(n \log n)$

문제 03 정답과 해설

$T(n)=T(n/2)+\Theta(n)$, $T(1)=\Theta(1)$, $n=2^k$ 일 때 차수는?

4번 $\Theta(n)$

해설

단계 수는 로그이지만 비용은 $n+n/2+n/4+\dots$ 이므로 선형이다.

문제 04

재귀 함수의 종료를 보장하는 설명으로 충분한 것은?

- 1 기저 사례가 있고 자연수 척도가 매 호출마다 엄격히 감소한다
- 2 호출할 때마다 인수가 증가하면 반드시 종료한다
- 3 입력의 절댓값이 감소하지 않아도 인수 하나가 바뀌면 된다
- 4 기저 사례가 있으면 그 조건에 도달하지 않아도 된다
- 5 매 호출에서 같은 크기의 새 배열을 만들면 된다

문제 04 정답과 해설

재귀 함수의 종료를 보장하는 설명으로 충분한 것은?

1번 기저 사례가 있고 자연수 척도가 매 호출마다 엄격히 감소한다

해설

기저 사례와 그 사례에 도달하는 감소 조건을 함께 확인해야 한다.

문제 05

$n!$ 을 n 부터 1까지 곱해 구한다. 재귀 구현은 n 을 1씩 줄이고 반복 구현도 n 회 안팎 곱한다. 기본 연산을 $O(1)$ 로 셀 때 옳은 설명은?

- 1 재귀 구현만 $\Theta(\log n)$ 시간이다
- 2 반복 구현은 항상 $\Theta(n^2)$ 시간이다
- 3 재귀 구현은 곱셈을 한 번만 한다
- 4 둘 다 $\Theta(n)$ 시간에 같은 값을 계산한다
- 5 두 구현은 서로 다른 팩토리얼 값을 계산한다

문제 05 정답과 해설

$n!$ 을 n 부터 1까지 곱해 구한다. 재귀 구현은 n 을 1씩 줄이고 반복 구현도 n 회 안팎 곱한다. 기본 연산을 $O(1)$ 로 셀 때 옳은 설명은?

4번 둘 다 $\Theta(n)$ 시간에 같은 값을 계산한다

해설

두 구현 모두 n 에 비례하는 횟수만큼 곱한다. 재귀를 썼다는 이유만으로 더 빠르거나 다른 결과가 나오지는 않는다.

문제 06

$F(0)=0$, $F(1)=1$ 인 순진한 피보나치 재귀가 느린 주된 이유는?

- 1 같은 인수의 부분 문제를 여러 번 계산한다
- 2 한 단계에서 두 값을 더하면 덧셈 횟수와 무관하게 지수 시간이 된다
- 3 $F(n-2)$ 는 $F(n-1)$ 의 계산에 포함되지 않으므로 독립이다
- 4 기저 사례가 두 개이면 호출 트리 깊이가 항상 2가 된다
- 5 재귀 호출 깊이만큼만 덧셈을 하므로 중복이 없다

문제 06 정답과 해설

$F(0)=0$, $F(1)=1$ 인 순진한 피보나치 재귀가 느린 주된 이유는?

1번 같은 인수의 부분 문제를 여러 번 계산한다

해설

호출 트리에 $F(k)$ 가 중복된다. 재귀 자체가 지수 시간을 뜻하지는 않는다.

문제 07

피보나치 반복에서 $a, b = b, a + b$ 를 수행할 때 올바른 해석은?

- 1 한 번에 두 항을 건너뛴다
- 2 a 와 b 가 항상 같은 값이 된다
- 3 b 만 변한다
- 4 a 를 먼저 바꾼 뒤 바뀐 a 로 b 를 계산한다
- 5 오른쪽의 두 식을 갱신 전 a, b 로 계산한다

문제 07 정답과 해설

피보나치 반복에서 $a, b = b, a + b$ 를 수행할 때 올바른 해석은?

5번 오른쪽의 두 식을 갱신 전 a, b 로 계산한다

해설

Python의 다중 대입은 오른쪽 값을 먼저 평가한다. C로 옮길 때는 임시 변수가 필요하다.

문제 08

한 번의 재귀 호출로 크기 $N/3$ 의 별무늬를 구해 재사용한다. 최종 $N \times N$ 문자를 한 글자씩 출력하는 작업 자체의 정확한 시간 차수는?

1 $\Theta(N^2)$

2 $\Theta(N \log N)$

3 $\Theta(\log N)$

4 $\Theta(N)$

5 $\Theta(1)$

문제 08 정답과 해설

한 번의 재귀 호출로 크기 $N/3$ 의 별무늬를 구해 재사용한다. 최종 $N \times N$ 문자를 한 글자씩 출력하는 작업 자체의 정확한 시간 차수는?

1번 $\Theta(N^2)$

해설

출력 문자 자체가 N^2 개이므로 깊이가 로그여도 전체 출력 시간은 최소 이차다.

문제 09

단계마다 후보를 하나 제거하는 순차 탐색의 실패 시 비교 횟수는? 후보 수는 n 이다.

1 $\lceil \log_2 n \rceil$

2 $n-1$

3 $n(n-1)/2$

4 n

5 1

문제 09 정답과 해설

단계마다 후보를 하나 제거하는 순차 탐색의 실패 시 비교 횟수는? 후보 수는 n 이다.

4번 n

해설

찾는 값이 없으면 모든 원소를 확인해야 한다.

문제 10

정렬되지 않은 배열에도 정확하게 적용할 수 있는 탐색은?

- 1 정렬 없이 제공된 간격의 값만 검사하는 탐색
- 2 순차 탐색
- 3 정렬 없이 중간 원소와 비교하여 한쪽을 버리는 이진 탐색
- 4 정렬 없이 끝값 비율로 위치를 추정해 한쪽을 버리는 보간 탐색
- 5 정렬 없이 첫 원소보다 작으면 즉시 실패하는 탐색

문제 10 정답과 해설

정렬되지 않은 배열에도 정확하게 적용할 수 있는 탐색은?

2번 순차 탐색

해설

순차 탐색은 순서에 의존하지 않고 각 원소를 직접 확인한다.

문제 11

이진 탐색에서 후보 구간을 $[lo, hi]$ 로 유지한다. $a[mid] < x$ 이면 올바른 갱신은?

1 $hi = mid$

2 $lo = mid$

3 $hi = mid - 1$

4 $lo = hi + 1$

5 $lo = mid + 1$

문제 11 정답과 해설

이진 탐색에서 후보 구간을 $[lo, hi]$ 로 유지한다. $a[mid] < x$ 이면 올바른 갱신은?

5번 $lo = mid + 1$

해설

mid 와 그 왼쪽은 모두 x 보다 작아 후보에서 제외된다.

문제 12

이진 탐색의 후보 구간이 $[lo, hi]$ 이고 $lo > hi$ 가 되었다. 의미는?

- 1 반드시 $a[lo]$ 가 정답이다
- 2 중간 인덱스를 한 번 더 검사해야 한다
- 3 배열이 비어 있었다는 뜻만 된다
- 4 찾는 값이 배열에 없다
- 5 원소가 중복된다는 뜻이다

문제 12 정답과 해설

이진 탐색의 후보 구간이 $[lo, hi]$ 이고 $lo > hi$ 가 되었다. 의미는?

4번 찾는 값이 배열에 없다

해설

유지한 후보 집합이 공집합이므로 실패로 종료한다.

문제 13

정렬된 배열에 중복값이 있을 때 일반적인 일치 즉시 반환 이진 탐색의 보장은?

- 1 항상 첫 인덱스를 반환한다
- 2 일치값의 개수를 반환한다
- 3 일치하는 인덱스 하나를 반환한다
- 4 중복값이 있으면 항상 실패한다
- 5 항상 마지막 인덱스를 반환한다

문제 13 정답과 해설

정렬된 배열에 중복값이 있을 때 일반적인 일치 즉시 반환 이진 탐색의 보장은?

3번 일치하는 인덱스 하나를 반환한다

해설

첫 위치를 원하면 일치 후에도 왼쪽에서 더 찾아야 한다.

문제 14

$\text{lower_bound}(x)$ 를 x 이상인 첫 원소의 인덱스로 정의한다. 모든 원소가 x 보다 작으면 반환값은?

1 $n-1$

2 배열 길이 n

3 -1

4 x

5 0

문제 14 정답과 해설

`lower_bound(x)`를 x 이상인 첫 원소의 인덱스로 정의한다. 모든 원소가 x 보다 작으면 반환값은?

2번 배열 길이 n

해설

삽입 위치는 n 까지 허용한다. 이는 검색 실패 표시 -1 과 다른 계약이다.

문제 15

고정 폭 정수에서 $(lo+hi)/2$ 의 덧셈 오버플로를 피하는 일반적인 식은? $0 \leq lo \leq hi$ 이다.

1 $(lo-hi)/2$

2 $lo+(hi-lo)/2$

3 $(hi-lo)/2$

4 $lo*hi/2$

5 $lo+hi/2$

문제 15 정답과 해설

고정 폭 정수에서 $(lo+hi)/2$ 의 덧셈 오버플로를 피하는 일반적인 식은? $0 \leq lo \leq hi$ 이다.

2번 $lo+(hi-lo)/2$

해설

차이는 구간 길이 이내이며 중간값은 lo 와 hi 사이에 남는다.

문제 16

보간 탐색의 위치 추정이 이진 탐색과 다른 점은?

- 1 정렬 여부를 무시한다
- 2 항상 가운데만 확인한다
- 3 끝점의 값과 목표값을 사용한다
- 4 항상 첫 원소만 확인한다
- 5 모든 후보의 합을 계산한다

문제 16 정답과 해설

보간 탐색의 위치 추정이 이진 탐색과 다른 점은?

3번 끝점의 값과 목표값을 사용한다

해설

보간 탐색은 값이 퍼진 정도를 이용해 예상 위치를 계산한다.

문제 17

보간 탐색에서 $a[l_o]=a[h_i]$ 일 때 가장 먼저 필요한 처리는?

- 1 목표값과 무관하게 실패를 반환한다
- 2 분모를 1로 바꿔 기존 위치 공식을 그대로 사용한다
- 3 목표값과 무관하게 l_o 를 반환한다
- 4 구간 길이가 1이 될 때까지 같은 나눗셈 공식을 반복한다
- 5 목표값과 비교해 성공 또는 실패로 종료한다

문제 17 정답과 해설

보간 탐색에서 $a[lo]=a[hi]$ 일 때 가장 먼저 필요한 처리는?

5번 목표값과 비교해 성공 또는 실패로 종료한다

해설

분모가 0이다. 정렬된 구간의 양 끝 값이 같으면 구간 전체 값도 같다.

문제 18

보간 탐색이 잘 맞는 입력의 특징은?

- 1 길이는 크지만 값 순서가 무작위인 배열이다
- 2 정렬되어 있고 값이 비교적 균등하게 분포한다
- 3 정렬되어 있지만 거의 모든 값이 한쪽 끝에 몰려 있다
- 4 인덱스 간격은 일정하지만 값의 분포는 고려하지 않는다
- 5 입력 길이가 2의 거듭제곱이면 값의 분포와 무관하다

문제 18 정답과 해설

보간 탐색이 잘 맞는 입력의 특징은?

2번 정렬되어 있고 값이 비교적 균등하게 분포한다

해설

평균적인 효율은 값 분포 가정에 영향을 받으며 최악 선형 시간 가능성은 남는다.

문제 19

보간 탐색의 최악 실행 시간에 대한 가장 정확한 점근적 차수는?

1 $\Theta(n)$

2 $\Theta(n^2)$

3 $\Theta(\log n)$

4 $\Theta(\log \log n)$

5 $\Theta(n \log n)$

문제 19 정답과 해설

보간 탐색의 최악 실행 시간에 대한 가장 정확한 점근적 차수는?

1번 $\Theta(n)$

해설

값 분포가 치우치면 조금씩만 후보가 줄어들 수 있다.

문제 20

정렬 비용 $\Theta(n \log n)$ 을 한 번 들인 뒤 이진 탐색을 m 번 수행한다. 총 시간은?

- 1 $\Theta(n \log m)$
- 2 $\Theta(m+n)$
- 3 $\Theta(nm \log n)$ 만 가능
- 4 $\Theta(n \log n + m \log n)$
- 5 $\Theta(\log(n+m))$

문제 20 정답과 해설

정렬 비용 $\Theta(n \log n)$ 을 한 번 들인 뒤 이진 탐색을 m 번 수행한다. 총 시간은?

4번 $\Theta(n \log n + m \log n)$

해설

전처리와 질의 비용을 더한다. 각 질의 때 다시 정렬할 필요가 없다.

문제 21

유클리드 알고리즘에서 $\gcd(a,b)=\gcd(b,a \bmod b)$ 가 성립하는 이유는?

- 1 두 수의 합이 보존되기 때문이다
- 2 두 쌍의 공약수 집합이 같다
- 3 나머지는 항상 1이기 때문이다
- 4 큰 수가 매번 정확히 절반이 되기 때문이다
- 5 두 수의 곱이 보존되기 때문이다

문제 21 정답과 해설

유클리드 알고리즘에서 $\gcd(a,b)=\gcd(b,a \bmod b)$ 가 성립하는 이유는?

2번 두 쌍의 공약수 집합이 같다

해설

$a=qb+r$ 관계를 이용하면 a,b 의 공약수와 b,r 의 공약수가 서로 일치한다.

문제 22

비음수 정수 a 에 대해 $\gcd(a,0)$ 의 반환값은?

- 1 $a-1$
- 2 1만 가능
- 3 정의할 수 없다
- 4 0만 가능
- 5 a

문제 22 정답과 해설

비음수 정수 a 에 대해 $\gcd(a, 0)$ 의 반환값은?

5번 a

해설

이 계약에서는 $\gcd(a, 0) = a$ 로 두며 $\gcd(0, 0) = 0$ 도 명시적으로 허용할 수 있다.

문제 23

유클리드 갱신 $(a,b) \leftarrow (b, a \bmod b)$ 에서 종료에 쓰는 감소 척도는? $b > 0$ 이다.

- 1 두 번째 인수 b
- 2 첫 번째 인수 a , 초기 순서와 무관하게 항상 감소
- 3 두 수의 공약수 개수
- 4 두 수의 최대공약수
- 5 두 수의 차 $a-b$, 항상 음이 아닌 채 감소

문제 23 정답과 해설

유클리드 갱신 $(a,b) \leftarrow (b, a \bmod b)$ 에서 종료에 쓰는 감소 척도는? $b > 0$ 이다.

1번 두 번째 인수 b

해설

새 두 번째 인수는 0 이상 b 미만이다.

문제 24

빠른 거듭제곱에서 $n=2k+1$ 이면 C^n 을 어떻게 구성하는가?

1 $(C^k)^2$

2 $C+(C^k)^2$

3 C^k+C^{k+1}

4 $2 \cdot C^k$

5 $C \cdot (C^k)^2$

문제 24 정답과 해설

빠른 거듭제곱에서 $n=2k+1$ 이면 C^n 을 어떻게 구성하는가?

5번 $C \cdot (C^k)^2$

해설

홀수 지수는 제곱 뒤 밑을 한 번 더 곱한다.

문제 25

$\text{fast}(C, n/2)$ 를 두 번 호출하는 구현과 한 번 호출해 결과를 재사용하는 구현의 차수는? 정수 연산 $O(1)$, $n=2^k$ 이다.

- 1 둘 다 $\Theta(\log n)$
- 2 각각 $\Theta(n^2)$, $\Theta(1)$
- 3 둘 다 $\Theta(n^2)$
- 4 각각 $\Theta(n)$, $\Theta(\log n)$
- 5 각각 $\Theta(\log n)$, $\Theta(n)$

문제 25 정답과 해설

fast(C,n/2)를 두 번 호출하는 구현과 한 번 호출해 결과를 재사용하는 구현의 차수는? 정수 연산 $O(1)$, $n=2^k$ 이다.

4번 각각 $\Theta(n)$, $\Theta(\log n)$

해설

두 호출이면 $T(n)=2T(n/2)+\Theta(1)$, 재사용하면 $T(n)=T(n/2)+\Theta(1)$ 이다.

문제 26

재귀 빠른 거듭제곱에서 $n \geq 2$ 이고, $n=0$ 까지 매번 $n//2$ 로 한 번 호출한다. 총 호출 횟수의 차수는?

1 $\Theta(n)$

2 $\Theta(\log n)$

3 $\Theta(n \log n)$

4 $\Theta(n^2)$

5 $\Theta(1)$

문제 26 정답과 해설

재귀 빠른 거듭제곱에서 $n \geq 2$ 이고, $n=0$ 까지 매번 $n//2$ 로 한 번 호출한다. 총 호출 횟수의 차수는?

2번 $\Theta(\log n)$

해설

매번 n 을 절반으로 줄이므로 0에 도달할 때까지의 총 호출 횟수는 $\Theta(\log n)$ 이다.

문제 27

삽입 정렬의 바깥 반복 i 직전에 유지하는 불변식은?

- 1 $a[i:n]$ 가 정렬되어 있다
- 2 $a[i]$ 가 전체 최솟값이다
- 3 $a[0:i]$ 가 정렬되어 있다
- 4 앞부분 원소가 모두 같은 값이다
- 5 전체 배열이 정렬되어 있다

문제 27 정답과 해설

삽입 정렬의 바깥 반복 i 직전에 유지하는 불변식은?

3번 $a[0:i]$ 가 정렬되어 있다

해설

정렬된 접두 구간에 현재 원소를 삽입하며 길이를 한 칸 늘린다.

문제 28

삽입 정렬의 안정성을 유지하는 이동 조건은? `key`를 오른쪽부터 삽입한다.

1 $a[j] \geq \text{key}$

2 $a[j] < \text{key}$

3 $a[j] \neq \text{key}$

4 $a[j] == \text{key}$

5 $a[j] > \text{key}$

문제 28 정답과 해설

삽입 정렬의 안정성을 유지하는 이동 조건은? key를 오른쪽부터 삽입한다.

5번 $a[j] > \text{key}$

해설

같은 키는 이동시키지 않아 기존의 상대 순서가 유지된다.

문제 29

이미 오름차순으로 정렬된 배열에 일반 삽입 정렬을 적용한 시간은?

- 1 $\Theta(n \log n)$
- 2 $\Theta(n^2)$ 가 반드시 필요하다
- 3 $\Theta(\log n)$
- 4 $\Theta(n)$
- 5 $\Theta(1)$

문제 29 정답과 해설

이미 오름차순으로 정렬된 배열에 일반 삽입 정렬을 적용한 시간은?

4번 $\Theta(n)$

해설

각 삽입에서 한 번 비교하고 이동하지 않는다. 바깥 반복은 $n-1$ 번이다.

문제 30

역순으로 정렬된 서로 다른 n 개 원소에서 삽입 정렬의 원소 이동 횟수는?

1 $n-1$

2 $\lceil \log_2 n \rceil$

3 $n(n-1)/2$

4 2^n

5 n

문제 30 정답과 해설

역순으로 정렬된 서로 다른 n 개 원소에서 삽입 정렬의 원소 이동 횟수는?

3번 $n(n-1)/2$

해설

i 번째 삽입이 i 개를 이동시키므로 1부터 $n-1$ 까지 합한다.

문제 31

가짜 동전이 정확히 하나이며 더 가볍다고 알려져 있다. 같은 개수 두 더미가 균형이면?

- 1 가짜는 왼쪽 더미에 있다
- 2 가짜는 저울에 올리지 않은 후보들 중에 있다
- 3 무게 정보로 아무것도 줄일 수 없다
- 4 가짜가 두 개다
- 5 가짜는 오른쪽 더미에 있다

문제 31 정답과 해설

가짜 동전이 정확히 하나이며 더 가볍다고 알려져 있다. 같은 개수 두 더미가 균형이면?

2번 가짜는 저울에 올리지 않은 후보들 중에 있다

해설

저울 위에 가짜가 있었다면 두 더미의 무게가 달라진다.

문제 32

동전 27개를 매번 세 그룹으로 나누어 한 번 비교한다. 가벼운 가짜 1개를 찾는 최악
저울질 횟수는?

1 3

2 4

3 2

4 9

5 27

문제 32 정답과 해설

동전 27개를 매번 세 그룹으로 나누어 한 번 비교한다. 가벼운 가짜 1개를 찾는 최악 저울질 횟수는?

1번 3

해설

27→9→3→1로 후보를 줄인다. 각 비교의 세 결과를 활용한다.

문제 33

동전 찾기에서 저울질 $O(\log n)$ 과 배열 합산 구현 $O(n)$ 이 함께 맞을 수 있는 이유는?

- 1 재귀 깊이가 로그이면 합산 작업을 포함한 시간도 로그다
- 2 배열 합은 크기와 무관하게 항상 상수 시간이다
- 3 실물 저울질도 후보 수만큼 반복하므로 선형이다
- 4 두 분석 모두 배열의 각 원소를 상수 번만 더한다는 뜻이다
- 5 세는 기본 연산이 다르다

문제 33 정답과 해설

동전 찾기에서 저울질 $O(\log n)$ 과 배열 합산 구현 $O(n)$ 이 함께 맞을 수 있는 이유는?

5번 세는 기본 연산이 다르다

해설

실물 저울의 한 번 비교와 배열 원소를 더하는 작업의 비용 모형을 구분해야 한다.

문제 34

퀵 셀렉트가 퀵 정렬과 달리 보통 수행하지 않는 작업은?

- 1 원소와 피벗 비교하기
- 2 피벗 기준으로 구분하기
- 3 정답이 없는 쪽을 재귀적으로 정렬하기
- 4 순위에 따라 다음 구간 정하기
- 5 기저 사례 검사하기

문제 34 정답과 해설

퀵 셀렉트가 퀵 정렬과 달리 보통 수행하지 않는 작업은?

3번 정답이 없는 쪽을 재귀적으로 정렬하기

해설

선택 문제는 한 순위의 값만 필요하므로 한쪽만 계속 처리한다.

문제 35

피벗보다 작은 값 L 개, 같은 값 E 개가 있다. k 가 1부터 시작할 때 피벗이 답인 조건은?

1 $k \geq L + E + 1$

2 $k > L + E$

3 $L < k \leq L + E$

4 $k = 0$

5 $k < L$

문제 35 정답과 해설

피벗보다 작은 값 L 개, 같은 값 E 개가 있다. k 가 1부터 시작할 때 피벗이 답인 조건은?

3번 $L < k \leq L + E$

해설

같은 값들이 정렬 결과의 $L+1$ 부터 $L+E$ 번째 순위를 차지한다.

문제 36

퀵 셀렉트에서 $k > L + E$ 라서 큰 값 구간으로 이동한다. 새 순위는?

1 $k - E$

2 $k - L - E$

3 $L + E - k$

4 $k - L$

5 $k + L + E$

문제 36 정답과 해설

퀵 선택트에서 $k > L + E$ 라서 큰 값 구간으로 이동한다. 새 순위는?

2번 $k - L - E$

해설

버린 작은 값과 같은 값의 개수를 모두 뺀다.

문제 37

항상 한 원소만 제거되는 퀵 셀렉트의 최악 시간은?

1 $\Theta(\log n)$

2 $\Theta(n)$

3 $\Theta(n^2)$

4 $\Theta(2^n)$

5 $\Theta(n \log n)$

문제 37 정답과 해설

항상 한 원소만 제거되는 퀵 셀렉트의 최악 시간은?

3번 $\Theta(n^2)$

해설

파티션 비용 $n+(n-1)+\dots$ 이 누적된다.

문제 38

서로 다른 원소에서 매번 균등한 무작위 피벗을 고르는 퀵 선택트의 시간에 대한 올바른 설명은?

- 1 기대 선형 시간이나 최악 이차 시간이다
- 2 기대 시간은 이차이고 최악 시간은 지수다
- 3 기대 시간은 로그이고 최악 시간은 선형이다
- 4 기대 시간과 최악 시간이 모두 $n \log n$ 이다
- 5 기대 시간과 최악 시간이 모두 선형이다

문제 38 정답과 해설

서로 다른 원소에서 매번 균등한 무작위 피벗을 고르는 퀵 선택트의 시간에 대한 올바른 설명은?

1번 기대 선형 시간이나 최악 이차 시간이다

해설

확률적 기대 성능과 모든 실행에 대한 최악 보장은 다르다.

문제 39

같은 값이 많은 입력에 세 갈래 파티션을 쓰는 주된 이점은?

- 1 무작위 피벗 없이도 모든 입력의 최악 선형 시간을 보장한다
- 2 서로 다른 값에 대해서도 매번 정확히 반으로 나눈다
- 3 같은 값의 원래 순서를 반드시 안정적으로 정렬한다
- 4 같은 값 구간의 순위들을 한 번에 처리한다
- 5 파티션 단계의 원소 비교 수를 항상 상수로 만든다

문제 39 정답과 해설

같은 값이 많은 입력에 세 갈래 파티션을 쓰는 주된 이점은?

4번 같은 값 구간의 순위들을 한 번에 처리한다

해설

같은 값 구간에 k 가 있으면 바로 답하고, 아니면 그 구간 전체를 제외한다.

문제 40

크기 n 의 배열을 반으로 자르는 재귀에서 매 단계 복사 비용 $\Theta(n)$ 이 든다.
 $T(n)=T(n/2)+\Theta(n)$ 의 전체 비용은?

1 $\Theta(n)$

2 $\Theta(n \log n)$

3 $\Theta(\log n)$

4 $\Theta(1)$

5 $\Theta(n^2)$

문제 40 정답과 해설

크기 n 의 배열을 반으로 자르는 재귀에서 매 단계 복사 비용 $\Theta(n)$ 이 든다. $T(n)=T(n/2)+\Theta(n)$ 의 전체 비용은?

1번 $\Theta(n)$

해설

호출 깊이만 세지 말고 복사되는 원소 수의 기하급수를 합해야 한다.

문제 41

$\text{fact}(0)=1$, $\text{fact}(n)=n \cdot \text{fact}(n-1)$ 이다. $\text{fact}(5)$ 의 값과 $\text{fact}(0)$ 을 포함한 총 호출 수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 41 정답과 해설

$\text{fact}(0)=1$, $\text{fact}(n)=n \cdot \text{fact}(n-1)$ 이다. $\text{fact}(5)$ 의 값과 $\text{fact}(0)$ 을 포함한 총 호출 수를 쓰시오.

120, 6회

해설

인수는 5,4,3,2,1,0이다. 곱셈은 5번이고 호출은 기저 사례까지 6번이다.

문제 42

$\text{fact}(n) = n \cdot \text{fact}(n-1)$, $\text{fact}(0) = 1$ 이다. $\text{fact}(4)$ 에서 $\text{fact}(0)$ 을 실행하는 순간
쌓인 호출 수와 첫 복귀값을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 42 정답과 해설

$\text{fact}(n)=n \cdot \text{fact}(n-1)$, $\text{fact}(0)=1$ 이다. $\text{fact}(4)$ 에서 $\text{fact}(0)$ 을 실행하는 순간 쌓인 호출 수와 첫 복귀값을 쓰시오.

5개, 1

해설

$\text{fact}(4)$, $\text{fact}(3)$, $\text{fact}(2)$, $\text{fact}(1)$, $\text{fact}(0)$ 이 쌓이고 기저 사례가 1을 반환한다.

문제 43

$a=0, b=1$ 에서 $a, b=b, a+b$ 를 6번 수행한다. 최종 (a, b) 를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 43 정답과 해설

$a=0, b=1$ 에서 $a, b=b, a+b$ 를 6번 수행한다. 최종 (a, b) 를 쓰시오.

(8,13)

해설

각 반복 후 (a, b) 는 연속된 피보나치 수이다. 6번 뒤 $F(6), F(7)$ 이다.

문제 44

순진한 $F(n)=F(n-1)+F(n-2)$, $F(0)=0, F(1)=1$ 구현에서 $F(4)$ 를 계산할 때 $F(2)$ 는 몇 번 호출되는가?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 44 정답과 해설

순진한 $F(n)=F(n-1)+F(n-2)$, $F(0)=0, F(1)=1$ 구현에서 $F(4)$ 를 계산할 때 $F(2)$ 는 몇 번 호출되는가?

2회

해설

$F(4)$ 의 왼쪽 $F(3)$ 안에 한 번, 오른쪽 자식으로 한 번 나온다.

문제 45

$F(0)=0, F(1)=1$ 이고 그 외에는 $F(n-1)+F(n-2)$ 를 호출하는 순진한 재귀다. $F(4)$ 의 총 함수 호출 수를 기저 사례까지 세시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 45 정답과 해설

$F(0)=0, F(1)=1$ 이고 그 외에는 $F(n-1)+F(n-2)$ 를 호출하는 순진한 재귀다. $F(4)$ 의 총 함수 호출 수를 기저 사례까지 세시오.

9회

해설

$C(0)=C(1)=1$, $C(n)=1+C(n-1)+C(n-2)$ 이므로 $C(2)=3, C(3)=5, C(4)=9$ 이다.

문제 46

배열 $[8, 3, 8, 2, 9]$ 를 왼쪽부터 순차 탐색해 첫 8의 인덱스를 반환한다. 반환값과 비교 횟수는? 인덱스는 0부터이다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 46 정답과 해설

배열 [8,3,8,2,9]를 왼쪽부터 순차 탐색해 첫 8의 인덱스를 반환한다. 반환값과 비교 횟수는?
인덱스는 0부터이다.

0, 1회

해설

첫 원소에서 즉시 일치하므로 뒤의 8은 확인하지 않는다.

문제 47

배열 $[8, 3, 8, 2, 9]$ 에서 7을 순차 탐색한다. 실패 시 -1 을 반환할 때 결과와 비교 횟수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 47 정답과 해설

배열 $[8, 3, 8, 2, 9]$ 에서 7을 순차 탐색한다. 실패 시 -1 을 반환할 때 결과와 비교 횟수는?

-1 , 5회

해설

모든 원소와 비교한 뒤 실패한다.

문제 48

배열 $[2, 5, 8, 11, 14, 17, 20]$ 에서 17을 이진 탐색한다. $[lo, hi]$, $mid = (lo + hi) // 2$ 를 사용한다. 검사한 mid 를 순서대로 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 48 정답과 해설

배열 [2,5,8,11,14,17,20]에서 17을 이진 탐색한다. [lo,hi], $\text{mid}=(\text{lo}+\text{hi})//2$ 를 사용한다. 검사한 mid를 순서대로 쓰시오.

3,5

해설

mid=3의 11보다 목표가 커서 [4,6]으로 줄고 mid=5에서 찾는다.

문제 49

배열 $[2, 5, 8, 11, 14, 17, 20]$ 에서 6을 이진 탐색한다. 닫힌 구간과 $\text{mid} = (\text{lo} + \text{hi}) // 2$ 를 쓴다. 검사 인덱스와 종료 시 (lo, hi) 를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 49 정답과 해설

배열 [2,5,8,11,14,17,20]에서 6을 이진 탐색한다. 닫힌 구간과 $\text{mid}=(\text{lo}+\text{hi})//2$ 를 쓴다. 검사 인덱스와 종료 시 (lo,hi)를 쓰시오.

3,1,2; (2,1)

해설

11보다 작아 [0,2], 5보다 커서 [2,2], 8보다 작아 $\text{hi}=1$ 이 된다.

문제 50

닫힌 구간 $[0,9]$ 의 가운데를 내림하여 선택한다. 가운데 값보다 목표가 크면 다음 후보
인덱스 구간과 원소 수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 50 정답과 해설

닫힌 구간 $[0,9]$ 의 가운데를 내림하여 선택한다. 가운데 값보다 목표가 크면 다음 후보 인덱스 구간과 원소 수는?

$[5,9]$, 5개

해설

$\text{mid}=4$ 이며 mid 를 제외한 오른쪽 다섯 원소만 남는다.

문제 51

[1,3,3,3,7]에서 `lower_bound(3)`와 `upper_bound(3)`를 쓰시오. `upper_bound`는 3보다 큰 첫 위치이다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 51 정답과 해설

[1,3,3,3,7]에서 lower_bound(3)와 upper_bound(3)를 쓰시오. upper_bound는 3보다 큰 첫 위치이다.

1,4

해설

같은 값 구간은 인덱스 [1,4)이므로 첫 위치 1, 끝 다음 위치 4이다.

문제 52

[1,3,3,3,7]에서 `lower_bound(8)`를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 52 정답과 해설

[1,3,3,3,7]에서 lower_bound(8)를 쓰시오.

5

해설

모든 값이 8보다 작으므로 배열 끝 다음 삽입 위치를 반환한다.

문제 53

서로 다른 15개 원소의 정렬 배열에서 닫힌 구간 이진 탐색을 한다. 최악의 반복 횟수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 53 정답과 해설

서로 다른 15개 원소의 정렬 배열에서 닫힌 구간 이진 탐색을 한다. 최악의 반복 횟수는?

4회

해설

후보 수 $15 \rightarrow 7 \rightarrow 3 \rightarrow 1$ 을 차례로 검사한다. 루프 한 번의 목표값 비교 묶음을 한 회로 센다.

문제 54

오름차순 배열 $a = [10, 20, 30, 40, 50]$ 에서 값 40을 찾으려고 한다.

인덱스는 0부터 시작하며, $a[i]$ 는 인덱스 i 에 있는 값이다.

탐색 구간의 시작 인덱스 $lo = 0$, 끝 인덱스 $hi = 4$, 찾을 값 $x = 40$ 이다.

보간 탐색에서 처음 확인할 인덱스 pos 를 다음 식으로 계산한다.

$$pos = lo + ((x - a[lo]) * (hi - lo)) // (a[hi] - a[lo])$$

여기서 $*$ 는 곱셈, $//$ 는 나눗셈의 정수 몫이다. 예: $7 // 2 = 3$.

위 식에 주어진 값을 대입하여 인덱스 pos 를 구하시오.

인덱스 값과, 값을 대입한 계산식 한 줄을 함께 쓰시오.

문제 54 정답과 해설

오름차순 배열 $a = [10, 20, 30, 40, 50]$ 에서 값 40을 찾으려고 한다.

인덱스는 0부터 시작하며, $a[i]$ 는 인덱스 i 에 있는 값이다.

탐색 구간의 시작 인덱스 $lo = 0$, 끝 인덱스 $hi = 4$, 찾을 값 $x = 40$ 이다.

보간 탐색에서 처음 확인할 인덱스 pos 를 다음 식으로 계산한다.

$$pos = lo + ((x - a[lo]) * (hi - lo)) // (a[hi] - a[lo])$$

여기서 $*$ 는 곱셈, $//$ 는 나눗셈의 정수 몫이다. 예: $7 // 2 = 3$.

위 식에 주어진 값을 대입하여 인덱스 pos 를 구하시오.

3

해설

$a[lo] = 10$, $a[hi] = 50$ 이므로

$$pos = 0 + ((40 - 10) * (4 - 0)) // (50 - 10)$$

$= 120 // 40 = 3$ 이다. 즉, 처음 확인할 곳은 인덱스 3(네 번째 원소)이다.

문제 55

$\text{gcd}(84, 30)$ 에서 b 가 0이 될 때까지 (a, b) 를 갱신한 결과 쌍들을 순서대로 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 55 정답과 해설

$\text{gcd}(84,30)$ 에서 b 가 0이 될 때까지 (a,b) 를 갱신한 결과 쌍들을 순서대로 쓰시오.

$(30,24) \rightarrow (24,6) \rightarrow (6,0)$

해설

$84 \bmod 30 = 24$, $30 \bmod 24 = 6$, $24 \bmod 6 = 0$ 이다. 최대공약수는 6이다.

문제 56

$\text{gcd}(35, 64)$ 를 유클리드 방식으로 계산한다. 첫 갱신과 최종 값을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 56 정답과 해설

$\text{gcd}(35, 64)$ 를 유클리드 방식으로 계산한다. 첫 갱신과 최종 값을 쓰시오.

(64, 35), 1

해설

$a < b$ 여도 $a \bmod b = a$ 이므로 먼저 자리가 바뀐다. 이후 64, 35, 29, 6, 5, 1, 0으로 진행한다.

문제 57

$\text{gcd}(a, 0) = a$ 를 사용한다. $\text{gcd}(0, 18)$ 에서 나머지 연산 횟수와 반환값은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 57 정답과 해설

$\text{gcd}(a,0)=a$ 를 사용한다. $\text{gcd}(0,18)$ 에서 나머지 연산 횟수와 반환값은?

1회, 18

해설

한 번 갱신하면 $(18,0)$ 이므로 종료한다.

문제 58

$\text{fast}(C, n)$ 가 $n=0$ 에서 1을 반환하고 그 외에는 $\text{fast}(C, n//2)$ 를 한 번 호출한다.
 $\text{fast}(2, 13)$ 의 인수 n 들을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 58 정답과 해설

$\text{fast}(C, n)$ 가 $n=0$ 에서 1을 반환하고 그 외에는 $\text{fast}(C, n//2)$ 를 한 번 호출한다. $\text{fast}(2, 13)$ 의 인수 n 들을 쓰시오.

$13 \rightarrow 6 \rightarrow 3 \rightarrow 1 \rightarrow 0$

해설

매번 정수 나눗셈으로 절반을 취한다. 총 호출은 5회이다.

문제 59

빠른 거듭제곱은 $n=0$ 에서 종료하고 그 외에는 $n//2$ 를 한 번 호출한다. 비기저 호출마다 제곱 1회, 홀수이면 곱셈 1회를 추가한다. $n=13$ 의 곱셈 횟수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 59 정답과 해설

빠른 거듭제곱은 $n=0$ 에서 종료하고 그 외에는 $n//2$ 를 한 번 호출한다. 비기저 호출마다 제곱 1회, 홀수이면 곱셈 1회를 추가한다. $n=13$ 의 곱셈 횟수는?

7회

해설

13,6,3,1의 제곱 4회와 홀수 13,3,1의 추가 곱셈 3회이다.

문제 60

$\text{fast}(C,n)$ 는 $n=0$ 에서 1을 반환한다. 그 외에는 $\text{fast}(C,n//2)$ 를 한 번 호출해 제공하고, 홀수이면 C 를 한 번 더 곱한다. $\text{fast}(3,5)$ 의 값과 곱셈 횟수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 60 정답과 해설

$\text{fast}(C, n)$ 는 $n=0$ 에서 1을 반환한다. 그 외에는 $\text{fast}(C, n//2)$ 를 한 번 호출해 제공하고, 홀수이면 C 를 한 번 더 곱한다. $\text{fast}(3, 5)$ 의 값과 곱셈 횟수는?

243, 5회

해설

지수 5, 2, 1에서 제공 3회, 홀수 두 곳에서 추가 2회이다.

문제 61

삽입 정렬로 $[5, 2, 4, 1]$ 을 정렬한다. 바깥 반복 $i=1$ 과 $i=2$ 직후 배열을 각각 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 61 정답과 해설

삽입 정렬로 [5,2,4,1]을 정렬한다. 바깥 반복 $i=1$ 과 $i=2$ 직후 배열을 각각 쓰시오.

[2,5,4,1], [2,4,5,1]

해설

2를 5 앞에, 다음 4를 2와 5 사이에 삽입한다.

문제 62

[4,3,2,1]에 삽입 정렬을 적용한다. 원소를 오른쪽으로 한 칸 미는 대입 횟수는? key를 넣는 대입은 제외한다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 62 정답과 해설

[4,3,2,1]에 삽입 정렬을 적용한다. 원소를 오른쪽으로 한 칸 미는 대입 횟수는? key를 넣는 대입은 제외한다.

6회

해설

각 삽입에서 1,2,3개를 이동시킨다.

문제 63

같은 키의 원래 순서를 A,B로 표시한 $[3A,1,3B,2]$ 를 안정적인 삽입 정렬로 정렬한 결과는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 63 정답과 해설

같은 키의 원래 순서를 A,B로 표시한 $[3A, 1, 3B, 2]$ 를 안정적인 삽입 정렬로 정렬한 결과는?

$[1, 2, 3A, 3B]$

해설

동일 키 3A와 3B의 상대 순서는 보존된다.

문제 64

가벼운 가짜 1개가 9개 동전 중에 있다. 첫 비교에서 1~3번과 4~6번이 균형이다. 다음 후보와 남은 최악 비교 횟수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 64 정답과 해설

가벼운 가짜 1개가 9개 동전 중에 있다. 첫 비교에서 1~3번과 4~6번이 균형이다. 다음 후보와 남은 최악 비교 횟수는?

7~9번, 1회

해설

7번과 8번을 비교하면 가벼운 쪽 또는 균형일 때 9번을 고를 수 있다.

문제 65

동전 81개를 세 그룹 방식으로 줄인다. 두 번의 비교 뒤 남을 수 있는 최대 후보 수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 65 정답과 해설

동전 81개를 세 그룹 방식으로 줄인다. 두 번의 비교 뒤 남을 수 있는 최대 후보 수는?

9개

해설

81→27→9이다. 각 단계에서 같은 크기 세 그룹으로 나눈다.

문제 66

배열 $[7, 2, 5, 2, 9, 4]$ 를 피벗 4로 세 갈래 분할한다. 원래 순서를 유지한 작은 값, 같은 값, 큰 값 구간을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 66 정답과 해설

배열 $[7, 2, 5, 2, 9, 4]$ 를 피벗 4로 세 갈래 분할한다. 원래 순서를 유지한 작은 값, 같은 값, 큰 값 구간을 쓰시오.

$[2, 2], [4], [7, 5, 9]$

해설

각 원소를 4와 비교하여 세 구간에 넣는다.

문제 67

[7,2,5,2,9,4]의 5번째 작은 값을 피벗 4의 세 갈래 파티션으로 찾는다. 큰 값 구간으로 이동할 때 새 순위와 최종 값을 쓰시오. 순위는 1부터다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 67 정답과 해설

[7,2,5,2,9,4]의 5번째 작은 값을 피벗 4의 세 갈래 파티션으로 찾는다. 큰 값 구간으로 이동할 때 새 순위와 최종 값을 쓰시오. 순위는 1부터다.

2번째, 7

해설

작은 값 2개와 같은 값 1개를 빼서 $k=2$ 가 된다. 큰 값 [7,5,9]의 두 번째 값은 7이다.

문제 68

[4,1,4,2,4,7]에서 4번째 작은 값을 피벗 4로 한 번 분할해 찾는다. L,E와 답을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 68 정답과 해설

[4,1,4,2,4,7]에서 4번째 작은 값을 피벗 4로 한 번 분할해 찾는다. L,E와 답을 쓰시오.

L=2,E=3, 답=4

해설

같은 값 구간이 3~5번째 순위를 차지한다.

문제 69

$T(1)=1$, $T(n)=T(n/2)+n$, $n=2^k$ 이다. $T(8)$ 의 정확한 값은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 69 정답과 해설

$T(1)=1$, $T(n)=T(n/2)+n$, $n=2^k$ 이다. $T(8)$ 의 정확한 값은?

15

해설

$T(8)=8+4+2+1=15$ 이다. 점근적 차수뿐 아니라 기저 비용을 포함한다.

문제 70

크기 27의 별무늬를 매번 크기의 $1/3$ 인 무늬 하나로 재귀 호출하여 만든다. 크기 1을 포함한 호출 깊이와 공백을 포함한 27×27 출력 문자 수는? 줄바꿈은 제외한다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 70 정답과 해설

크기 27의 별무늬를 매번 크기의 1/3인 무늬 하나로 재귀 호출하여 만든다. 크기 1을 포함한 호출 깊이와 공백을 포함한 27×27 출력 문자 수는? 줄바꿈은 제외한다.

4단계, 729개

해설

크기 27→9→3→1이며 최종 27×27의 모든 위치를 출력한다. 줄바꿈은 문자 수에서 제외한다.

문제 71

반으로 줄이는 알고리즘은 항상 $O(\log n)$ 이라는 주장에 반례 점화식 하나와 분석을 제시하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 71 정답과 해설

반으로 줄이는 알고리즘은 항상 $O(\log n)$ 이라는 주장에 반례 점화식 하나와 분석을 제시하시오.

$T(n)=T(n/2)+n$ 은 $\Theta(n)$ 이다.

해설

한 단계의 작업이 상수가 아니다. $n+n/2+\dots < 2n$ 이고 첫 단계만으로도 n 이 필요하다.

문제 72

단힌 구간 이진 탐색에서 $a[mid] < x$ 일 때 $lo = mid$ 로 갱신하면 종료하지 않을 수 있다.
최소 길이 반례를 제시하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 72 정답과 해설

닫힌 구간 이진 탐색에서 $a[mid] < x$ 일 때 $lo = mid$ 로 갱신하면 종료하지 않을 수 있다. 최소 길이 반례를 제시하시오.

$a=[1,2]$, $x=2$, $lo=0, hi=1$ 이면 $mid=0$ 이 계속된다.

해설

lo 가 그대로 0에 남는다. 이미 검사한 mid 를 제외하는 $lo = mid + 1$ 로 고쳐야 한다.

문제 73

일반 이진 탐색으로 첫 등장 위치를 구할 수 없는 이유와 수정 방향을 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 73 정답과 해설

일반 이진 탐색으로 첫 등장 위치를 구할 수 없는 이유와 수정 방향을 설명하시오.

일치한 mid 왼쪽에도 같은 값이 있을 수 있다. 답 후보를 저장하고 $hi=mid-1$ 로 계속 찾는다.

해설

[2,2,2]에서 $mid=1$ 을 즉시 반환하면 첫 위치 0을 놓친다. lower_bound를 구한 뒤 일치 여부를 확인해도 된다.

문제 74

정렬되지 않은 $[4, 1, 3]$ 에서 일반 이진 탐색으로 4를 찾을 때 실패하는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 74 정답과 해설

정렬되지 않은 $[4, 1, 3]$ 에서 일반 이진 탐색으로 4를 찾을 때 실패하는 이유를 설명하시오.

mid=1의 값 1을 보고 왼쪽을 버리면 실제 4도 버린다.

해설

절반을 버리는 근거는 정렬 순서이다. 중간값 비교만으로 무정렬 구간의 나머지 값을 판단할 수 없다.

문제 75

배열 원소가 모두 5인 경우 보간 탐색의 예외 처리와 정확성을 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 75 정답과 해설

배열 원소가 모두 5인 경우 보간 탐색의 예외 처리와 정확성을 설명하시오.

양 끝 값이 같으면 $x=5$ 일 때 성공, 그 외에는 실패로 종료한다.

해설

정렬된 배열에서 양 끝이 같으면 내부도 같다. 분모 0인 추정식은 사용하지 않는다.

문제 76

많은 탐색 질의가 있는 고정 배열에서 정렬 후 이진 탐색을 고려하는 이유를 총 비용으로 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 76 정답과 해설

많은 탐색 질의가 있는 고정 배열에서 정렬 후 이진 탐색을 고려하는 이유를 총 비용으로 설명하시오.

순차 탐색 m 번은 최악 $\Theta(mn)$, 한 번 정렬 후 이진 탐색은 $\Theta(n \log n + m \log n)$ 이다.

해설

정렬 비용을 여러 질의에 나누어 부담한다. 질의 수가 적으면 전처리 없는 탐색이 나올 수 있다.

문제 77

유클리드 알고리즘은 (a,b) 를 $(b,a\%b)$ 로 바꾸고 $b=0$ 이면 멈춘다. $(18,12)$ 에서 시작할 때 순서대로 바뀌는 쌍과 최종 최대공약수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 77 정답과 해설

유클리드 알고리즘은 (a,b) 를 $(b,a\%b)$ 로 바꾸고 $b=0$ 이면 멈춘다. $(18,12)$ 에서 시작할 때 순서대로 바뀌는 쌍과 최종 최대공약수를 쓰시오.

$(18,12) \rightarrow (12,6) \rightarrow (6,0)$, 최대공약수 6

해설

$18\%12=6$, $12\%6=0$ 이다. 두 번째 값이 0이 되었을 때 첫 번째 값 6이 최대공약수다.

문제 78

빠른 거듭제곱에서 재귀 결과를 변수에 저장하지 않고 같은 호출을 두 번 하는 것이 왜 문제인가?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 78 정답과 해설

빠른 거듭제곱에서 재귀 결과를 변수에 저장하지 않고 같은 호출을 두 번 하는 것이 왜 문제인가?

같은 $C^{(n//2)}$ 를 중복 계산하여 호출 수가 늘어난다.

해설

한 번만 계산하면 $\Theta(\log n)$, 두 번 계산하면 $n=2^k$ 에서 $\Theta(n)$ 이다. 결과를 저장한 뒤 제공해야 한다.

문제 79

다음 C 코드는 피보나치 수열의 다음 두 항을 계산하려는 코드이다.
갱신 전 a 는 현재 항, b 는 다음 항이다. 갱신 후에는 각각
기존 b 와 기존 $a + b$ 가 되어야 한다.

```
a = b;  
b = a + b;
```

- ① 갱신 전 $a = 2$, $b = 3$ 일 때, 위 코드 실행 후 a 와 b 의 값을 쓰시오.
- ② 의도한 결과와 달라지는 이유를 설명하고, 임시 변수를 사용하여 수정하시오.

배점: 실행 결과 2점 · 오류 원인 3점 · 수정안 5점. C 코드로 수정안을 쓰시오.

문제 79 정답과 해설

$a = 2, b = 3$ 에서 순차 대입의 결과를 확인하고 수정한다.

① 실제 결과: $a = 3, b = 6$ / 의도한 결과: $a = 3, b = 5$

해설

② $a = b;$ 를 실행하면 기존 a 의 값 2가 사라진다.
따라서 $b = a + b;$ 는 $2 + 3$ 이 아니라 $3 + 3$ 을 계산한다.

```
int t = a + b;  
a = b;  
b = t;
```

덮어쓰기 전에 합 5를 t 에 저장하므로, 갱신 후 $a = 3, b = 5$ 가 된다.

문제 80

삽입 정렬의 안정성을 $a[j] > \text{key}$ 조건과 연결해 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 80 정답과 해설

삽입 정렬의 안정성을 $a[j] > \text{key}$ 조건과 연결해 설명하시오.

같은 키는 오른쪽으로 이동하지 않으므로 새 key는 기존 같은 키 뒤에 놓인다.

해설

비교를 $\geq$ 로 바꾸면 앞서 나온 같은 키를 넘어갈 수 있어 상대 순서가 뒤집힌다.

문제 81

삽입 정렬의 최선 시간과 최악 시간이 다른 이유를 원소 이동 횟수로 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 81 정답과 해설

삽입 정렬의 최선 시간과 최악 시간이 다른 이유를 원소 이동 횟수로 설명하시오.

정렬 입력은 이동 0회이고 각 단계 비교 후 끝난다. 역순 입력은 $1+\dots+(n-1)$ 회 이동한다.

해설

바깥 반복 횟수는 같아도 내부 반복 길이가 달라 최선 $\Theta(n)$, 최악 $\Theta(n^2)$ 이다.

문제 82

가짜가 더 가볍다는 조건을 제거하면 기존 동전 찾기 논리를 그대로 쓸 수 없는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 82 정답과 해설

가짜가 더 가볍다는 조건을 제거하면 기존 동전 찾기 논리를 그대로 쓸 수 없는 이유를 설명하시오.

기울었을 때 가벼운 쪽의 가벼운 가짜인지 무거운 쪽의 무거운 가짜인지 구분할 수 없다.

해설

기존 전략은 가벼운 쪽만 후보로 남긴다. 가짜 무게 방향이 미지이면 후보 상태에 방향도 포함해야 한다.

문제 83

겉모양이 같은 동전 27개 중 하나만 더 가볍다. 후보를 같은 수의 세 그룹으로 나누고 두 그룹을 양팔저울로 비교해 가벼운 동전이 있는 그룹만 남긴다. 후보가 1개가 될 때까지 몇 회 비교하는지 후보 수와 함께 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 83 정답과 해설

겉모양이 같은 동전 27개 중 하나만 더 가볍다. 후보를 같은 수의 세 그룹으로 나누고 두 그룹을 양팔저울로 비교해 가벼운 동전이 있는 그룹만 남긴다. 후보가 1개가 될 때까지 몇 회 비교하는지 후보 수와 함께 쓰시오.

27 → 9 → 3 → 1, 총 3회

해설

저울이 기울면 가벼운 쪽, 같으면 올리지 않은 그룹에 가짜가 있다. 한 번마다 후보가 3분의 1로 줄어 3회면 1개가 남는다.

문제 84

동전 배열의 절반 합을 직접 계산하는 구현이 저울 횟수와 다른 시간 차수를 갖는 이유를 비용 합으로 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 84 정답과 해설

동전 배열의 절반 합을 직접 계산하는 구현이 저울 횟수와 다른 시간 차수를 갖는 이유를 비용 합으로 쓰시오.

합산 원소 수가 $n+n/2+n/4+\dots$ 이므로 $\Theta(n)$ 이다.

해설

비교 횟수만 세면 $\Theta(\log n)$ 이지만 각 비교를 위한 합산 비용은 입력 크기에 비례한다.

문제 85

퀵 선택트에서 큰 값 구간의 새 순위를 $k-L$ 이 아닌 $k-L-E$ 로 정해야 하는 이유를 반례로 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 85 정답과 해설

퀵 선택트에서 큰 값 구간의 새 순위를 $k-L$ 이 아닌 $k-L-E$ 로 정해야 하는 이유를 반례로 설명하시오.

[1,2,2,2,5], 피벗 2, $k=5$ 이면 $L=1, E=3$ 이다. 큰 구간 [5]에서는 1번째를 찾아야 한다.

해설

$k-L=4$ 는 크기 1인 구간에서 유효하지 않다. 같은 값 E 개도 앞선 순위에 포함된다.

문제 86

퀵 선택트의 기대 $O(n)$ 이 모든 실행의 $O(n)$ 을 뜻하지 않는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 86 정답과 해설

퀵 셀렉트의 기대 $O(n)$ 이 모든 실행의 $O(n)$ 을 뜻하지 않는 이유를 설명하시오.

무작위 피벗이 반복해서 극단값이면 후보가 한 개씩만 줄어 이차 시간이 걸린다.

해설

기대값은 피벗 선택의 확률을 평균한 비용이다. 최악 보장과 평균 성능은 별도로 표현한다.

문제 87

정렬 배열에서 x 의 출현 횟수를 $O(\log n)$ 에 구하는 설계를 제시하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 87 정답과 해설

정렬 배열에서 x 의 출현 횟수를 $O(\log n)$ 에 구하는 설계를 제시하시오.

처음 x 이상인 위치 L 과 처음 x 보다 큰 위치 U 를 이진 탐색으로 구해 $U-L$ 을 반환한다.

해설

x 가 없으면 두 경계가 같아 0이다. 경계의 반환 범위는 0부터 n 까지이다.

문제 88

한 단계에 $n-1$ 크기로 줄어드는 사실만으로 $O(n)$ 이라 할 수 없는 이유를 삽입 정렬로 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 88 정답과 해설

한 단계에 $n-1$ 크기로 줄어드는 사실만으로 $O(n)$ 이라 할 수 없는 이유를 삽입 정렬로 설명하시오.

각 단계에서 새 원소를 삽입하는 비용이 최대 $\Theta(n)$ 이므로 전체는 $\Theta(n^2)$ 까지 된다.

해설

단계 수와 단계당 비용을 곱하거나 합해야 한다. 입력이 이미 정렬되었을 때에만 각 삽입이 상수 시간이다.

문제 89

재귀 함수가 입력을 줄이지만 0에서 다시 자기 자신을 부른다. 왜 기저 사례가 필요한가?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 89 정답과 해설

재귀 함수가 입력을 줄이지만 0에서 다시 자기 자신을 부른다. 왜 기저 사례가 필요한가?

감소할 수 없는 최소 입력에서는 답을 바로 반환해야 한다.

해설

예를 들어 $n//2$ 는 $n=0$ 에서 계속 0이다. 감소식만으로 종료가 보장되지 않으므로 호출 전에 기저 사례를 검사한다.

문제 90

3^k 크기의 별무늬에서 재귀 깊이와 출력 크기를 구분해 복잡도를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 90 정답과 해설

3^k 크기의 별무늬에서 재귀 깊이와 출력 크기를 구분해 복잡도를 설명하시오.

깊이는 $k+1=\Theta(\log N)$, 출력은 N^2 개 위치이므로 시간 하한 $\Omega(N^2)$ 이다.

해설

하위 패턴을 한 번 재사용하더라도 최종 문자열들을 만드는 작업과 출력 작업은 남는다.

문제 91

정수 배열 a 에서 x 가 처음 등장하는 0 기반 인덱스를 반환한다. 없으면 -1 이다.
 $0 \leq n \leq 1000$. 함수 이름은 `first_index`다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 91 정답과 해설

첫 등장 위치

왼쪽부터 검사하고 첫 일치에서 반환한다.

해설

시간 $O(n)$.

문제 91 Python 구현

첫 등장 위치

```
01 def first_index(a, x):  
02     for i, value in enumerate(a):  
03         if value == x:  
04             return i  
05     return -1
```

시간 $O(n)$.

문제 91 C 구현

첫 등장 위치

```
01  int first_index(const int a[], int n, int x) {  
02      for (int i = 0; i < n; ++i)  
03          if (a[i] == x) return i;  
04      return -1;  
05  }
```

시간 $O(n)$.

문제 92

서로 다른 정수가 오름차순인 배열 a 에서 x 의 0 기반 인덱스를 반환한다. 없으면 -1 이다. $0 \leq n \leq 1000$. `binary_search`를 작성한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 92 정답과 해설

이진 탐색

양 끝을 포함한 탐색 구간을 유지하며 중간값 비교 후 구간을 줄인다.

해설

시간 $O(\log(n+1))$.

문제 92 Python 구현

이진 탐색

```
01 def binary_search(a, x):
02     lo, hi = 0, len(a) - 1
03     while lo <= hi:
04         mid = lo + (hi - lo) // 2
05         if a[mid] == x:
06             return mid
07         if a[mid] < x:
08             lo = mid + 1
09         else:
10             hi = mid - 1
11     return -1
```

시간 $O(\log(n+1))$.

문제 92 C 구현

이진 탐색

```
01  int binary_search(const int a[], int n, int x) {
02      int lo = 0, hi = n - 1;
03      while (lo <= hi) {
04          int mid = lo + (hi - lo) / 2;
05          if (a[mid] == x) return mid;
06          if (a[mid] < x) lo = mid + 1;
07          else hi = mid - 1;
08      }
09      return -1;
10  }
```

시간 $O(\log(n+1))$.

문제 93

오름차순 배열 a 에서 $a[i] \geq x$ 인 첫 인덱스를 반환한다. 없으면 n 이다. 중복을 허용하고 $0 \leq n \leq 1000$ 이다. `lower_bound`를 작성한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 93 정답과 해설

하한 위치

반열린 구간 $[lo, hi)$ 에서 답의 경계를 좁힌다.

해설

시간 $O(\log(n+1))$.

문제 93 Python 구현

하한 위치

```
01 def lower_bound(a, x):
02     lo, hi = 0, len(a)
03     while lo < hi:
04         mid = (lo + hi) // 2
05         if a[mid] < x:
06             lo = mid + 1
07         else:
08             hi = mid
09     return lo
```

시간 $O(\log(n+1))$.

문제 93 C 구현

하한 위치

```
01  int lower_bound(const int a[], int n, int x) {  
02      int lo = 0, hi = n;  
03      while (lo < hi) {  
04          int mid = lo + (hi - lo) / 2;  
05          if (a[mid] < x) lo = mid + 1;  
06          else hi = mid;  
07      }  
08      return lo;  
09  }
```

시간 $O(\log(n+1))$.

문제 94

오름차순 배열 a 에서 x 의 등장 횟수를 로그 시간에 반환하는 `count_equal`을 작성한다.
 $0 \leq n \leq 1000$. x 이상인 첫 위치(없으면 n)를 주는 `lower_bound`가 제공된다. Python
또는 C로 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 94 정답과 해설

중복값 개수

x보다 큰 첫 위치에서 x 이상인 첫 위치를 뺀다.

해설

시간 $O(\log(n+1))$. $x+1$ 을 쓰지 않아 최댓값 오버플로를 피한다.

문제 94 Python 구현

중복값 개수

```
01 def count_equal(a, x):
02     lo, hi = 0, len(a)
03     while lo < hi:
04         mid = (lo + hi) // 2
05         if a[mid] <= x:
06             lo = mid + 1
07         else:
08             hi = mid
09     return lo - lower_bound(a, x)
```

시간 $O(\log(n+1))$. $x+1$ 을 쓰지 않아 최댓값 오버플로를 피한다.

문제 94 C 구현

중복값 개수

```
01  /* lower_bound(a, n, x) is provided. */
02  int count_equal(const int a[], int n, int x) {
03      int lo = 0, hi = n;
04      while (lo < hi) {
05          int mid = lo + (hi - lo) / 2;
06          if (a[mid] <= x) lo = mid + 1;
07          else hi = mid;
08      }
09      return lo - lower_bound(a, n, x);
10  }
```

시간 $O(\log(n+1))$. $x+1$ 을 쓰지 않아 최댓값 오버플로를 피한다.

문제 95

$0 \leq a, b \leq 10^9$ 이며 둘 다 0인 경우도 허용한다. `gcd_value(a,b)`를 반복문으로 작성한다. `gcd(0,0)`은 0으로 정의한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 95 정답과 해설

유클리드 호제법

(a,b) 를 $(b, a \bmod b)$ 로 바꾸다가 b 가 0이면 a 를 반환한다.

해설

시간 $O(\log(\max(a,b)+2))$.

문제 95 Python 구현

유클리드 호제법

```
01 def gcd_value(a, b):  
02     while b != 0:  
03         a, b = b, a % b  
04     return a
```

시간 $O(\log(\max(a,b)+2))$.

문제 95 C 구현

유클리드 호제법

```
01  int gcd_value(int a, int b) {  
02      while (b != 0) {  
03          int r = a % b;  
04          a = b;  
05          b = r;  
06      }  
07      return a;  
08  }
```

시간 $O(\log(\max(a,b)+2))$.

문제 96

$\text{power_mod}(a,n,m)$ 은 $a^n \bmod m$ 을 반환한다. $0 \leq a \leq 10^9$, $0 \leq n \leq 10^9$, $1 \leq m \leq 10^9$.
 $n=0$ 이면 $1 \bmod m$ 이다. C는 long long을 사용한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 96 정답과 해설

분할 거듭제곱

지수를 반으로 줄이며 홀수 비트에 해당하는 밑만 답에 곱한다.

해설

시간 $O(\log(n+1))$. 중간 곱은 10^{18} 미만으로 signed long long 범위 안이다.

문제 96 Python 구현

분할 거듭제곱

```
01 def power_mod(a, n, m):
02     result = 1 % m
03     a %= m
04     while n > 0:
05         if n % 2:
06             result = result * a % m
07             a = a * a % m
08             n //= 2
09     return result
```

시간 $O(\log(n+1))$. 중간 곱은 10^{18} 미만으로 signed long long 범위 안이다.

문제 96 C 구현

분할 거듭제곱

```
01 long long power_mod(long long a, long long n,  
02                     long long m) {  
03     long long result = 1 % m;  
04     a %= m;  
05     while (n > 0) {  
06         if (n % 2) result = result * a % m;  
07         a = a * a % m;  
08         n /= 2;  
09     }  
10     return result;  
11 }
```

시간 $O(\log(n+1))$. 중간 곱은 10^{18} 미만으로 signed long long 범위 안이다.

문제 97

정수 배열 a 를 제자리 오름차순으로 정렬하는 `insertion_sort`를 작성한다.
 $0 \leq n \leq 1000$. 같은 값은 앞에 있던 원소를 뒤로 밀지 않는다. Python 또는 C 중 하나로
함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 97 정답과 해설

안정적인 삽입 정렬

현재 값을 보관한 뒤 그보다 큰 앞 원소만 한 칸씩 민다.

해설

최악 시간 $O(n^2)$, 정렬된 입력 $O(n)$.

문제 97 Python 구현

안정적인 삽입 정렬

```
01 def insertion_sort(a):
02     for i in range(1, len(a)):
03         key = a[i]
04         j = i - 1
05         while j >= 0 and a[j] > key:
06             a[j + 1] = a[j]
07             j -= 1
08         a[j + 1] = key
```

최악 시간 $O(n^2)$, 정렬된 입력 $O(n)$.

문제 97 C 구현

안정적인 삽입 정렬

```
01 void insertion_sort(int a[], int n) {  
02     for (int i = 1; i < n; ++i) {  
03         int key = a[i], j = i - 1;  
04         while (j >= 0 && a[j] > key) {  
05             a[j + 1] = a[j];  
06             --j;  
07         }  
08         a[j + 1] = key;  
09     }  
10 }
```

최악 시간 $O(n^2)$, 정렬된 입력 $O(n)$.

문제 98

$1 \leq n \leq 1000$ 인 배열에서 중복도 각각 세어 k 번째 작은 값을 반환한다. k 는 1부터 n 까지다. `quick_select`를 작성하며 배열 변경을 허용한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 98 정답과 해설

퀵 선택

피벗 이하를 앞으로 모으고 피벗의 최종 위치와 목표 인덱스를 비교한다.

해설

이 고정 피벗 구현의 최악 시간 $O(n^2)$. 모든 입력에서 선형이라고 주장할 수 없다.

문제 98 Python 구현

퀵 선택

```
01 def quick_select(a, k):
02     lo, hi, target = 0, len(a) - 1, k - 1
03     while lo <= hi:
04         pivot, p = a[hi], lo
05         for j in range(lo, hi):
06             if a[j] <= pivot:
07                 a[p], a[j] = a[j], a[p]
08                 p += 1
09         a[p], a[hi] = a[hi], a[p]
10         if p == target:
11             return a[p]
12         if p < target:
13             lo = p + 1
14         else:
15             hi = p - 1
```

이 고정 피벗 구현의 최악 시간 $O(n^2)$. 모든 입력에서 선형이라고 주장할 수 없다.

문제 98 C 구현

퀵 선택

```
01  int quick_select(int a[], int n, int k) {
02      int lo=0, hi=n-1, target=k-1;
03      while (lo <= hi) {
04          int pivot=a[hi], p=lo;
05          for (int j=lo; j<hi; ++j) {
06              if (a[j] <= pivot) {
07                  int t=a[p]; a[p++]=a[j]; a[j]=t;
08              }
09          }
10          int t=a[p]; a[p]=a[hi]; a[hi]=t;
11          if (p == target) return a[p];
12          if (p < target) lo=p+1;
13          else hi=p-1;
14      }
15      return 0; /* unreachable for valid k */
16  }
```

이 고정 피벗 구현의 최악 시간 $O(n^2)$. 모든 입력에서 선형이라고 주장할 수 없다.

문제 99

$F(0)=0, F(1)=1$ 이다. $0 \leq n \leq 90$ 에서 $\text{fibonacci}(n)$ 을 반복문으로 작성한다. C의 반환형은 long long이며 다음 항을 불필요하게 계산하지 않는다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 99 정답과 해설

반복 피보나치

$n \geq 2$ 에서 $F(2)$ 부터 $F(n)$ 까지만 차례로 계산한다.

해설

시간 $O(n)$. $F(90)=2880067194370816120$ 이다.

문제 99 Python 구현

반복 피보나치

```
01 def fibonacci(n):
02     if n < 2:
03         return n
04     a, b = 0, 1
05     for i in range(2, n + 1):
06         a, b = b, a + b
07     return b
```

시간 $O(n)$. $F(90)=2880067194370816120$ 이다.

문제 99 C 구현

반복 피보나치

```
01 long long fibonacci(int n) {  
02     if (n < 2) return n;  
03     long long a=0, b=1;  
04     for (int i=2; i<=n; ++i) {  
05         long long next=a+b;  
06         a=b; b=next;  
07     }  
08     return b;  
09 }
```

시간 $O(n)$. $F(90)=2880067194370816120$ 이다.

문제 100

길이 $n=2^k$ ($1 \leq n \leq 1024$)인 `weights`에 0이 정확히 하나 있고 나머지는 1이다. 왼쪽 절반 합을 조사하며 0의 인덱스를 찾는 `light_coin`을 작성한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 100 정답과 해설

가벼운 동전 찾기

왼쪽 합이 원소 수보다 작으면 왼쪽, 같으면 오른쪽으로 범위를 줄인다.

해설

배열 합 연산을 포함한 시간 $O(n)$. 범위를 줄이는 단계 수만 $O(\log n)$ 이다.

문제 100 Python 구현

가벼운 동전 찾기

```
01 def light_coin(weights):
02     lo, hi = 0, len(weights)
03     while hi - lo > 1:
04         mid = (lo + hi) // 2
05         total = sum(weights[i] for i in range(lo, mid))
06         if total < mid - lo:
07             hi = mid
08         else:
09             lo = mid
10     return lo
```

배열 합 연산을 포함한 시간 $O(n)$. 범위를 줄이는 단계 수만 $O(\log n)$ 이다.

문제 100 C 구현

가벼운 동전 찾기

```
01  int light_coin(const int weights[], int n) {
02      int lo=0, hi=n;
03      while (hi-lo > 1) {
04          int mid=(lo+hi)/2, total=0;
05          for (int i=lo; i<mid; ++i) total+=weights[i];
06          if (total < mid-lo) hi=mid;
07          else lo=mid;
08      }
09      return lo;
10  }
```

배열 합 연산을 포함한 시간 $O(n)$. 범위를 줄이는 단계 수만 $O(\log n)$ 이다.