문제 01

분할 정복의 세 단계 순서로 옳은 것은?

1. 부분 문제 해결→분할→결합
2. 결합→분할→부분 문제 해결
3. 분할→부분 문제 해결→결합
4. 부분 문제 해결→결합→분할
5. 분할→결합→부분 문제 해결

문제 01 정답과 해설

분할 정복의 세 단계 순서로 옳은 것은?

3번 분할→부분 문제 해결→결합

해설

작은 문제를 해결한 결과를 이용해 원래 문제의 답을 구성한다.

문제 02

합병 정렬의 분할 뒤 반드시 필요한 전제는?

1. 모든 원소가 서로 다르다
2. 입력이 이미 정렬되어 있다
3. 두 부분을 각각 정렬한 뒤 병합한다
4. 두 부분의 길이가 항상 같다
5. 두 부분의 값 범위가 겹치지 않는다

문제 02 정답과 해설

합병 정렬의 분할 뒤 반드시 필요한 전제는?

3번 두 부분을 각각 정렬한 뒤 병합한다

해설

병합은 정렬된 두 배열을 입력으로 받는다. 중복과 홀수 길이도 처리할 수 있다.

문제 03

길이 p,q인 정렬 배열의 표준 병합에서 최악 원소 비교 횟수는? $p,q \geq 1$이다.

1 $\lceil \log_2(p+q) \rceil$

2 $\min(p,q)-1$

3 $p+q$

4 pq

5 $p+q-1$

문제 03 정답과 해설

길이 p,q인 정렬 배열의 표준 병합에서 최악 원소 비교 횟수는? p,q≥1이다.

5번 p+q−1

해설

한쪽이 먼저 소진되면 남은 원소는 비교 없이 복사한다. 마지막 하나를 남길 때까지 비교할 수 있다.

문제 04

안정적인 병합 정렬에서 양쪽 현재 키가 같을 때 먼저 꺼낼 원소는?

1. 같은 키의 대표 하나만 남긴다
2. 오른쪽 원소를 먼저 꺼내고 끝에서 전체를 뒤집는다
3. 두 원소 중 삽입 위치가 더 뒤인 쪽을 먼저 쓴다
4. 동률을 임의로 처리해도 안정성이 항상 유지된다
5. 왼쪽 부분의 원소

문제 04 정답과 해설

안정적인 병합 정렬에서 양쪽 현재 키가 같을 때 먼저 꺼낼 원소는?

5번 왼쪽 부분의 원소

해설

원래 배열에서 왼쪽 부분이 앞에 있으므로 동등 키의 상대 순서를 보존한다.

문제 05

일반 합병 정렬의 점화식은? 분할 크기는 절반, 병합은 선형이다.

1 $T(n)=T(n/2)+\Theta(1)$

2 $T(n)=4T(n/2)+\Theta(n^2)$

3 $T(n)=T(n-1)+\Theta(1)$

4 $T(n)=2T(n-1)+\Theta(1)$

5 $T(n)=2T(n/2)+\Theta(n)$

문제 05 정답과 해설

일반 합병 정렬의 점화식은? 분할 크기는 절반, 병합은 선형이다.

5번 $T(n)=2T(n/2)+\Theta(n)$

해설

두 부분을 모두 재귀적으로 정렬하고 전체 길이에 비례해 병합한다.

문제 06

이미 정렬된 입력에도 병합을 생략하지 않는 일반 합병 정렬의 시간 차수는?

1. $\Theta(1)$
2. $\Theta(n^2)$
3. $\Theta(n \log n)$
4. $\Theta(\log n)$
5. $\Theta(n)$

문제 06 정답과 해설

이미 정렬된 입력에도 병합을 생략하지 않는 일반 합병 정렬의 시간 차수는?

3번 Θ(n log n)

해설

모든 재귀 단계에서 각 원소를 병합하므로 최선 입력에서도 같은 점근적 비용이다.

문제 07

총 n개(n≥2)의 원소를 가진 두 정렬 배열을 한 번 병합한다. 최악의 원소 비교 횟수의 차수는?

1 $\Theta(n^2)$

2 $\Theta(n \log n)$가 반드시 필요

3 $\Theta(1)$

4 $\Theta(\log n)$만 필요

5 $\Theta(n)$

문제 07 정답과 해설

총 n개(n≥2)의 원소를 가진 두 정렬 배열을 한 번 병합한다. 최악의 원소 비교 횟수의 차수는?

5번 Θ(n)

해설

한 번의 비교마다 원소 하나가 확정된다. 최악 n−1번 비교하므로 Θ(n)이다.

문제 08

길이 0 또는 1인 배열이 합병 정렬의 기저 사례인 이유는?

1. 빈 배열에서는 비교할 값이 없어 반드시 오류가 나야 한다
2. 두 원소가 같을 때만 기저 사례로 처리할 수 있다
3. 길이 1에서도 병합용 두 비어 있지 않은 배열이 필요하다
4. 길이가 1이면 추가로 자기 자신을 정렬해야 한다
5. 이미 정렬되어 있고 더 나눌 필요가 없다

문제 08 정답과 해설

길이 0 또는 1인 배열이 합병 정렬의 기저 사례인 이유는?

5번 이미 정렬되어 있고 더 나눌 필요가 없다

해설

빈 배열도 정렬된 배열로 취급하면 경계 처리가 자연스럽다.

문제 09

비어 있지 않은 최대 연속 부분배열을 찾는다. 모든 값이 음수라면 정답은?

1. 0
2. 가장 큰 원소 하나
3. 양 끝 원소의 합
4. 전체 배열의 합
5. 가장 작은 원소 하나

문제 09 정답과 해설

비어 있지 않은 최대 연속 부분배열을 찾는다. 모든 값이 음수라면 정답은?

2번 가장 큰 원소 하나

해설

빈 구간을 허용하지 않으므로 0으로 초기화하면 잘못된 답이 될 수 있다.

문제 10

분할 정복 최대 연속합에서 고려해야 할 경우는?

1. 왼쪽 내부와 오른쪽 내부만 비교한다
2. 양쪽 전체 합과 각 끝 원소만 비교한다
3. 경계를 가로지르는 구간만 비교한다
4. 길이가 정확히 n/2인 연속 구간만 비교한다
5. 왼쪽 내부·오른쪽 내부·경계를 가로지르는 구간

문제 10 정답과 해설

분할 정복 최대 연속합에서 고려해야 할 경우는?

5번 왼쪽 내부·오른쪽 내부·경계를 가로지르는 구간

해설

모든 연속 구간은 이 세 경우 중 하나에 속한다.

문제 11

경계를 가로지르는 최대 연속합의 올바른 계산은?

1. 왼쪽 정답과 오른쪽 정답 중 작은 값
2. 왼쪽 최대 접두 합+오른쪽 최대 접미 합
3. 왼쪽 전체 최댓값+오른쪽 전체 최댓값
4. 왼쪽 최대 접미 합+오른쪽 최대 접두 합
5. 양쪽 전체 합의 최솟값

문제 11 정답과 해설

경계를 가로지르는 최대 연속합의 올바른 계산은?

4번 왼쪽 최대 접미 합+오른쪽 최대 접두 합

해설

경계를 반드시 포함해야 하므로 왼쪽 끝부분과 오른쪽 앞부분이 이어져야 한다.

문제 12

최대 교차합 계산에서 왼쪽과 오른쪽 스캔은 각각 어디서 시작하는가? 닫힌 구간을 mid에서 나눈다.

1. lo와 hi
2. hi와 mid+1
3. mid−1과 mid
4. mid와 mid+1
5. lo와 mid

문제 12 정답과 해설

최대 교차합 계산에서 왼쪽과 오른쪽 스캔은 각각 어디서 시작하는가? 닫힌 구간을 mid에서 나눈다.

4번 mid와 mid+1

해설

왼쪽 부분은 mid를 포함하고 오른쪽 부분은 mid+1에서 시작한다.

문제 13

교차합을 매번 선형 스캔하는 분할 정복 최대 연속합의 시간은?

1. $\Theta(n^2 \log n)$
2. $\Theta(n \log n)$
3. $\Theta(n^2)$
4. $\Theta(n)$
5. $\Theta(\log n)$

문제 13 정답과 해설

교차합을 매번 선형 스캔하는 분할 정복 최대 연속합의 시간은?

2번 Θ(n log n)

해설

두 절반의 답과 선형 결합으로 $T(n)=2T(n/2)+\Theta(n)$이다.

문제 14

최대 연속합과 임의 부분집합의 최대 합이 다른 이유는?

1. 선택한 원소의 개수를 항상 n/2로 제한한다
2. 첫 원소와 마지막 원소를 반드시 함께 선택한다
3. 선택 전에 원소를 오름차순으로 재배열한다
4. 양수 원소만 입력받는다
5. 선택한 인덱스가 연속이어야 한다

문제 14 정답과 해설

최대 연속합과 임의 부분집합의 최대 합이 다른 이유는?

5번 선택한 인덱스가 연속이어야 한다

해설

중간 음수도 구간의 일부로 포함해야 한다. 양수만 골라 더하는 문제와 다르다.

문제 15

퀵 정렬에서 a=[8,4,6,3,7,1]의 가운데 값 6을 피벗으로 정한다.
분할 전에 피벗과 맨 끝 값을 교환한 직후의 배열은?

1 [8,4,1,3,7,6]

2 [6,4,8,3,7,1]

3 [1,3,4,6,7,8]

4 [8,4,6,3,1,7]

5 [1,4,6,3,7,8]

문제 15 정답과 해설

퀵 정렬에서 a=[8,4,6,3,7,1]의 가운데 값 6을 피벗으로 정한다.
분할 전에 피벗과 맨 끝 값을 교환한 직후의 배열은?

1번 [8,4,1,3,7,6]

해설

가운데 인덱스는 2, 맨 끝 인덱스는 5이다.
두 칸의 값 6과 1만 교환하므로 [8,4,1,3,7,6]이 된다.

문제 16

a=[2,7,1,6,5]에서 피벗은 맨 끝의 5, L=0, R=3이다.
L≤R이고 a[L]≤5인 동안 L을 늘린 뒤,
L≤R이고 a[R]>5인 동안 R을 줄인다. 멈춘 L, R은?

1 L=0, R=3

2 L=1, R=2

3 L=2, R=1

4 L=4, R=0

5 L=1, R=3

문제 16 정답과 해설

a=[2,7,1,6,5]에서 피벗은 맨 끝의 5, L=0, R=3이다.
L≤R이고 a[L]≤5인 동안 L을 늘린 뒤,
L≤R이고 a[R]>5인 동안 R을 줄인다. 멈춘 L, R은?

2번 L=1, R=2

해설

L은 2를 지나 7에서 멈춰 1이다.
R은 6을 지나 1에서 멈춰 2이다.

문제 17

a=[2,7,1,6,5]에서 피벗은 5이고 L=1, R=2에서 멈췄다.
L≤R이므로 a[L]과 a[R]을 교환한다. 교환 직후의 배열은?

1 [2,7,5,6,1]

2 [1,2,7,6,5]

3 [2,1,7,6,5]

4 [2,5,1,6,7]

5 [1,2,5,6,7]

문제 17 정답과 해설

a=[2,7,1,6,5]에서 피벗은 5이고 L=1, R=2에서 멈췄다.
L≤R이므로 a[L]과 a[R]을 교환한다. 교환 직후의 배열은?

3번 [2,1,7,6,5]

해설

현재 a[1]=7은 오른쪽으로, a[2]=1은 왼쪽으로 보낸다.
두 값만 교환하면 [2,1,7,6,5]이다. 전체 정렬은 아직 끝나지 않았다.

문제 18

퀵 정렬의 분할은 피벗 이하의 값을 모두 왼쪽으로 보낸다.
입력 n개가 모두 같은 값이면 매번 n−1개인 왼쪽 구간만 남는다.
이 경우 전체 시간 복잡도는?

1 O(1)

2 O(log n)

3 $O(n^2)$

4 O(n)

5 O(n log n)

문제 18 정답과 해설

퀵 정렬의 분할은 피벗 이하의 값을 모두 왼쪽으로 보낸다.
입력 n개가 모두 같은 값이면 매번 n−1개인 왼쪽 구간만 남는다.
이 경우 전체 시간 복잡도는?

3번 $O(n^2)$

해설

남는 길이가 n−1, n−2, …로 줄어드는 긴 재귀 사슬이다.
각 구간의 선형 분할 비용을 합하면 $O(n^2)$이다.

문제 19

빠른 거듭제곱은 half=mod_power(a,b//2,m)을 한 번 구한 뒤
(half * half)%m을 계산한다. half에 결과를 저장하는 이유는?

1 지수를 무조건 0으로 바꾸기 위해

2 같은 절반 지수의 결과를 재귀로 두 번 계산하지 않기 위해

3 배열을 정렬하기 위해

4 m을 절반으로 줄이기 위해

5 나머지 연산을 없애기 위해

문제 19 정답과 해설

빠른 거듭제곱은 half=mod_power(a,b//2,m)을 한 번 구한 뒤
(half * half)%m을 계산한다. half에 결과를 저장하는 이유는?

2번 같은 절반 지수의 결과를 재귀로 두 번 계산하지 않기 위해

해설

절반 지수의 결과를 한 번만 구하고 그 숫자를 재사용한다.
같은 작은 문제를 두 번 호출할 필요가 없다.

문제 20

쿼드 트리의 현재 영역은 시작 행 r=2, 시작 열 c=4, 한 변 n=4이다.
네 등분할 때 우상 영역의 (시작 행, 시작 열, 한 변)은?

1 (4,4,2)

2 (2,4,2)

3 (4,6,2)

4 (2,6,2)

5 (2,8,4)

문제 20 정답과 해설

쿼드 트리의 현재 영역은 시작 행 r=2, 시작 열 c=4, 한 변 n=4이다.
네 등분할 때 우상 영역의 (시작 행, 시작 열, 한 변)은?

4번 (2,6,2)

해설

우상은 아래로 이동하지 않고 오른쪽으로 half=2칸 이동한다.
따라서 행은 2, 열은 4+2=6, 한 변은 2이다.

문제 21

빠른 거듭제곱 함수는 지수 b가 0이 아니면 b//2를 다음 지수로 쓴다.
현재 b=13일 때 바로 다음 재귀 호출의 지수는? //는 정수 나눗셈이다.

1	26
2	12
3	6.5
4	7
5	6

문제 21 정답과 해설

빠른 거듭제곱 함수는 지수 b가 0이 아니면 b//2를 다음 지수로 쓴다.
현재 b=13일 때 바로 다음 재귀 호출의 지수는? //는 정수 나눗셈이다.

5번 6

해설

13//2는 소수 부분을 버린 정수 6이다.
다음 호출에서는 지수 6을 처리한다.

문제 22

흑백 쿼드트리에서 현재 영역이 한 색이면 올바른 처리는?

1. 그 색 하나를 기록하고 종료한다
2. 색 대신 영역의 넓이만 기록한다
3. 색 대신 검은 칸의 개수만 기록한다
4. 반드시 1×1이 될 때까지 나눈다
5. 같은 색이어도 네 자식의 기록을 모두 저장한다

문제 22 정답과 해설

흑백 쿼드트리에서 현재 영역이 한 색이면 올바른 처리는?

1번 그 색 하나를 기록하고 종료한다

해설

균일 영역을 리프 하나로 표현하는 것이 압축의 핵심이다.

문제 23

쿼드트리에서 혼합된 정사각형을 나누는 자식 수는?

1. 2
2. 9
3. 3
4. 4
5. 8

문제 23 정답과 해설

쿼드트리에서 혼합된 정사각형을 나누는 자식 수는?

4번 4

해설

좌상·우상·좌하·우하의 네 사분면으로 나눈다.

문제 24

쿼드트리 부호화 순서를 좌상·우상·좌하·우하로 정했다. 복호화에 이 순서가 필요한 이유는?

1. 복호화 시 자식을 색 값으로 다시 정렬해야 한다
2. 부호 길이만 같으면 방문 순서는 달라도 된다
3. 각 부분의 공간 위치를 복원해야 하기 때문이다
4. 색이 두 가지이면 사분면의 순서는 정보에 영향을 주지 않는다
5. 자식들의 색 빈도만 알면 위치는 자동으로 복원된다

문제 24 정답과 해설

쿼드트리 부호화 순서를 좌상·우상·좌하·우하로 정했다. 복호화에 이 순서가 필요한 이유는?

3번 각 부분의 공간 위치를 복원해야 하기 때문이다

해설

색 목록만 같아도 자식 순서가 달라지면 복원 영상이 달라진다.

문제 25

흑백 쿼드트리 압축 효과가 가장 큰 예는?

1. 끝까지 분할해야 하는 영상
2. 1칸마다 색이 바뀌는 큰 체커보드
3. 전체가 한 색인 영상
4. 모든 리프가 1×1이 되는 큰 영상
5. 모든 2×2 블록이 혼합된 영상

문제 25 정답과 해설

흑백 쿼드트리 압축 효과가 가장 큰 예는?

3번 전체가 한 색인 영상

해설

전체 한 색이면 크기와 무관하게 색 기호 하나로 표현할 수 있다.

문제 26

쿼드트리를 이용한 충돌 후보 탐색에서 주의할 점은?

1. 모든 객체는 점 하나와 완전히 같다
2. 경계를 걸친 객체와 이웃 영역을 누락하지 않는다
3. 후보만 같으면 정밀 검사는 필요 없다
4. 객체 크기는 고려하지 않아도 된다
5. 다른 리프 객체는 절대로 충돌하지 않는다

문제 26 정답과 해설

쿼드트리를 이용한 충돌 후보 탐색에서 주의할 점은?

2번 경계를 걸친 객체와 이웃 영역을 누락하지 않는다

해설

공간 분할은 후보를 줄이는 장치다. 실제 영역이 겹치는 객체는 경계 양쪽에도 있을 수 있다.

문제 27

마스터 정리 $T(n)=aT(n/b)+f(n)$에서 a의 의미는?

1. 재귀 부분 문제의 개수
2. 부분 문제 크기를 나누는 비율
3. 각 기저 사례에 남는 입력 크기
4. 재귀 트리의 전체 깊이
5. 재귀 호출 외 결합 비용의 지수

문제 27 정답과 해설

마스터 정리 T(n)=aT(n/b)+f(n)에서 a의 의미는?

1번 재귀 부분 문제의 개수

해설

b는 크기 축소 비율, f(n)은 재귀 호출 외 작업 비용이다.

문제 28

$T(n)=4T(n/2)+\Theta(n)$, $T(1)=\Theta(1)$의 차수는?

1. $\Theta(n)$
2. $\Theta(n^2)$
3. $\Theta(\log n)$
4. $\Theta(n^2 \log n)$
5. $\Theta(n \log n)$

문제 28 정답과 해설

$T(n)=4T(n/2)+\Theta(n)$, $T(1)=\Theta(1)$의 차수는?

2번 $\Theta(n^2)$

해설

기준 지수 $\log_2 4=2$이고 결합 비용은 그보다 다항식 차이로 작다.

문제 29

$T(n)=2T(n/2)+\Theta(n^2)$의 차수는?

1. $\Theta(n)$
2. $\Theta(\log n)$
3. $\Theta(n \log n)$
4. $\Theta(n^2)$
5. $\Theta(n^3)$

문제 29 정답과 해설

$T(n)=2T(n/2)+\Theta(n^2)$의 차수는?

4번 $\Theta(n^2)$

해설

각 아래 층의 전체 비용이 절반으로 줄어 최상위 층이 지배한다.

문제 30

$T(n)=3T(n/2)+\Theta(n)$의 차수는?

1. $\Theta(n^{(\log_2 3)})$
2. $\Theta(\log n)$
3. $\Theta(n^3)$
4. $\Theta(n \log n)$
5. $\Theta(n^2)$

문제 30 정답과 해설

$T(n)=3T(n/2)+\Theta(n)$의 차수는?

1번 $\Theta(n^{\log_2 3})$

해설

기준 지수 $\log_2 3 \approx 1.585$가 결합 비용의 지수 1보다 크다.

문제 31

$T(n)=T(n-1)+n$에 기본 마스터 정리를 바로 적용할 수 없는 이유는?

1. 부분 문제 크기가 n/b 꼴의 고정 비율이 아니다
2. 재귀 호출 개수가 반드시 2 이상이어야 하기 때문이다
3. a와 b가 같은 값이어야 하기 때문이다
4. $T(1)$이 상수이면 정리를 쓸 수 없기 때문이다
5. 결합 비용이 선형이면 정리를 쓸 수 없기 때문이다

문제 31 정답과 해설

$T(n)=T(n-1)+n$에 기본 마스터 정리를 바로 적용할 수 없는 이유는?

1번 부분 문제 크기가 n/b 꼴의 고정 비율이 아니다

해설

한 칸 감소 점화식은 대입이나 합으로 분석한다.

문제 32

n은 2의 거듭제곱이다. 병합 정렬에서 각 층의 처리량은 $\Theta(n)$, 층수는 $\Theta(\log_2 n)$이다. 전체 시간 복잡도는?

1. $\Theta(n)$
2. $\Theta(n \log n)$
3. $\Theta(\log n)$
4. $\Theta(n^2)$
5. $\Theta(1)$

문제 32 정답과 해설

n은 2의 거듭제곱이다. 병합 정렬에서 각 층의 처리량은 Θ(n), 층수는 Θ($\log_2 n$)이다. 전체 시간 복잡도는?

2번 Θ(n log n)

해설

층마다 n에 비례하는 작업을 $\log_2 n$에 비례하는 횟수만큼 하므로 Θ(n log n)이다.

문제 33

정렬된 A=[1,4], B=[2,3]을 오름차순으로 병합한다.
완성된 배열은?

1. [1,4,2,3]
2. [1,2,3,4]
3. [2,3,1,4]
4. [4,3,2,1]
5. [1,2,4,3]

문제 33 정답과 해설

정렬된 A=[1,4], B=[2,3]을 오름차순으로 병합한다.
완성된 배열은?

2번 [1,2,3,4]

해설

두 배열의 맨 앞에서 남아 있는 값 중 작은 값을 고른다.
1, 2, 3, 4 순서로 결과에 넣는다.

문제 34

A=[1,4], B=[2,5]를 병합하는 중 i=1, j=0이다.
A[i]와 B[j] 중 다음에 넣을 값과 갱신할 인덱스는?

1 4를 넣고 i를 2로 바꾼다

2 2를 넣고 i를 2로 바꾼다

3 4를 넣고 j를 1로 바꾼다

4 2를 넣고 j를 1로 바꾼다

5 2를 넣고 i와 j를 모두 늘린다

문제 34 정답과 해설

A=[1,4], B=[2,5]를 병합하는 중 i=1, j=0이다.
A[i]와 B[j] 중 다음에 넣을 값과 갱신할 인덱스는?

4번 2를 넣고 j를 1로 바꾼다

해설

현재 값은 A[1]=4와 B[0]=2이다. 작은 값 2를 넣는다.
B에서 값을 꺼냈으므로 j만 0에서 1로 늘린다.

문제 35

병합 중 A의 원소는 모두 사용했고 B에는 [3,7]이 남았다.
남은 원소를 처리하는 올바른 방법은?

1. 3과 7을 그 순서대로 결과 뒤에 붙인다
2. 7과 3을 그 순서대로 결과 뒤에 붙인다
3. 남은 두 원소를 버린다
4. 3만 넣고 7은 버린다
5. 병합을 처음부터 다시 시작한다

문제 35 정답과 해설

병합 중 A의 원소는 모두 사용했고 B에는 [3,7]이 남았다.
남은 원소를 처리하는 올바른 방법은?

1번 3과 7을 그 순서대로 결과 뒤에 붙인다

해설

B 자체가 정렬되어 있으므로 남은 3, 7의 순서를 유지한다.
반대쪽 배열이 비었으므로 두 배열 사이의 비교는 더 필요 없다.

문제 36

A=[2,6], B=[1,5]를 i=0, j=0부터 병합한다.
1, 2를 차례로 결과에 넣은 직후의 i, j는?

1 i=2, j=0

2 i=0, j=2

3 i=0, j=1

4 i=1, j=0

5 i=1, j=1

문제 36 정답과 해설

A=[2,6], B=[1,5]를 i=0, j=0부터 병합한다.
1, 2를 차례로 결과에 넣은 직후의 i, j는?

5번 i=1, j=1

해설

1은 B에서 꺼냈으므로 j가 1이 된다.
2는 A에서 꺼냈으므로 i도 1이 된다.

문제 37

퀵 정렬에서 a=[8,4,6,3,7,1], lo=0, hi=5이다.
가운데 인덱스 (lo+hi)//2의 값을 피벗으로 고르면 피벗 값은?

1 8

2 4

3 6

4 3

5 1

문제 37 정답과 해설

퀵 정렬에서 a=[8,4,6,3,7,1], lo=0, hi=5이다.
가운데 인덱스 (lo+hi)//2의 값을 피벗으로 고르면 피벗 값은?

3번 6

해설

가운데 인덱스는 (0+5)//2=2이고 a[2]=6이다.
인덱스 2와 그 위치의 값 6을 구분한다.

문제 38

피벗은 a[5]=6이고 a=[3,4,1,8,7,6], L=3이다.
마지막에 a[L]과 a[5]를 교환하면 피벗의 인덱스는?

1 5

2 4

3 2

4 3

5 0

문제 38 정답과 해설

피벗은 a[5]=6이고 a=[3,4,1,8,7,6], L=3이다.
마지막에 a[L]과 a[5]를 교환하면 피벗의 인덱스는?

4번 3

해설

마지막 교환으로 피벗 6이 L이 가리킨 자리로 이동한다.
L=3이므로 확정된 피벗 인덱스는 3이다.

문제 39

퀵 정렬의 [2,7] 구간에서 피벗 인덱스 p=4가 확정되었다.
피벗을 제외하고 재귀 정렬할 두 구간은? 양 끝을 포함한다.

1. [2,4]와 [4,7]
2. [2,3]와 [5,7]
3. [2,4]와 [5,7]
4. [2,3]와 [4,7]
5. [2,7]과 [2,7]

문제 39 정답과 해설

퀵 정렬의 [2,7] 구간에서 피벗 인덱스 p=4가 확정되었다.
피벗을 제외하고 재귀 정렬할 두 구간은? 양 끝을 포함한다.

2번 [2,3]와 [5,7]

해설

피벗 p=4는 이미 제자리를 찾았다.
왼쪽은 [lo,p-1]=[2,3], 오른쪽은 [p+1,hi]=[5,7]이다.

문제 40

퀵 정렬의 포인터 탐색에서 배열 값을 읽기 전에
L≤R인지 먼저 확인하는 주된 이유는?

1 현재 탐색 구간 밖의 값을 읽지 않기 위해

2 모든 원소를 같은 값으로 만들기 위해

3 배열 길이를 늘리기 위해

4 피벗 값을 매번 바꾸기 위해

5 재귀 호출을 모두 없애기 위해

문제 40 정답과 해설

퀵 정렬의 포인터 탐색에서 배열 값을 읽기 전에
L≤R인지 먼저 확인하는 주된 이유는?

1번 현재 탐색 구간 밖의 값을 읽지 않기 위해

해설

포인터가 서로 지나친 뒤에는 현재 탐색 구간이 비어 있다.
L≤R을 먼저 검사하면 구간 밖의 값을 읽지 않고 탐색을 끝낸다.

문제 41

[1,4,7]과 [2,3,8]을 표준 병합한다. 결과와 원소 비교 횟수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 41 정답과 해설

[1,4,7]과 [2,3,8]을 표준 병합한다. 결과와 원소 비교 횟수를 쓰시오.

[1,2,3,4,7,8], 5회

해설

1:2,4:2,4:3,4:8,7:8을 비교한 뒤 남은 8을 복사한다.

문제 42

[1,2]와 [3,4,5]를 병합한다. 양쪽 현재 값의 비교 횟수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 42 정답과 해설

[1,2]와 [3,4,5]를 병합한다. 양쪽 현재 값의 비교 횟수는?

2회

해설

1과 3,2와 3을 비교한 뒤 오른쪽 나머지를 복사한다.

문제 43

[8,3,6,1]을 합병 정렬한다. 가장 아래의 두 병합 결과와 최종 결과를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 43 정답과 해설

[8,3,6,1]을 합병 정렬한다. 가장 아래의 두 병합 결과와 최종 결과를 쓰시오.

[3,8], [1,6]; [1,3,6,8]

해설

각 두 원소를 먼저 정렬한 뒤 두 정렬 배열을 병합한다.

문제 44

길이 8의 배열을 한 원소까지 반으로 나눈다. 리프 수와 내부 병합 호출 수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 44 정답과 해설

길이 8의 배열을 한 원소까지 반으로 나눈다. 리프 수와 내부 병합 호출 수는?

8개, 7회

해설

리프 8개인 완전 이진 재귀 트리의 내부 노드는 7개이다.

문제 45

T(1)=0, T(n)=2T(n/2)+n이다. T(8)을 구하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 45 정답과 해설

T(1)=0, T(n)=2T(n/2)+n이다. T(8)을 구하시오.

24

해설

각 병합 층 비용이 8이고 병합 층이 3개이다.

문제 46

[−4,7,−2,5,−9,3]의 비어 있지 않은 최대 연속합과 해당 구간을 쓰시오. 인덱스는 0부터이다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 46 정답과 해설

[−4,7,−2,5,−9,3]의 비어 있지 않은 최대 연속합과 해당 구간을 쓰시오. 인덱스는 0부터이다.

10, [1,3]

해설

7−2+5=10이며 양옆의 −4나 −9를 포함하면 줄어든다.

문제 47

[−8,−3,−5]의 비어 있지 않은 최대 연속합과 구간을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 47 정답과 해설

[−8,−3,−5]의 비어 있지 않은 최대 연속합과 구간을 쓰시오.

−3, [1,1]

해설

가장 덜 작은 원소 하나를 고른다. 빈 구간 0은 허용되지 않는다.

문제 48

[4,−6,3,5,−2,1]을 인덱스 [0,2]와 [3,5]로 나눈다. 왼쪽 최대 접미 합, 오른쪽 최대 접두 합, 교차합은? 각 구간은 비어 있지 않다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 48 정답과 해설

[4,−6,3,5,−2,1]을 인덱스 [0,2]와 [3,5]로 나눈다. 왼쪽 최대 접미 합, 오른쪽 최대 접두 합, 교차합은? 각 구간은 비어 있지 않다.

3,5,8

해설

왼쪽은 [3]이 3, 오른쪽은 [5]가 5로 최대이다. 경계 연결 구간은 [3,5]이다.

문제 49

[5,−100,6]에서 양수만 더한 값과 실제 최대 연속합을 각각 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 49 정답과 해설

[5,−100,6]에서 양수만 더한 값과 실제 최대 연속합을 각각 쓰시오.

11,6

해설

5와 6을 함께 포함하려면 −100도 포함해야 한다.

문제 50

길이 6인 배열의 비어 있지 않은 연속 구간 개수를 구하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 50 정답과 해설

길이 6인 배열의 비어 있지 않은 연속 구간 개수를 구하시오.

21개

해설

길이별 6+5+4+3+2+1=n(n+1)/2이다.

문제 51

[2,−1,2,−1]의 전체 합, 최대 접두 합, 최대 접미 합, 최대 연속합을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 51 정답과 해설

[2,−1,2,−1]의 전체 합, 최대 접두 합, 최대 접미 합, 최대 연속합을 쓰시오.

2,3,2,3

해설

접두 최대는 처음 3개, 접미 최대는 전체, 연속 최대는 인덱스 0~2이다.

문제 52

a=[2,7,1,6,5]에서 피벗은 맨 끝의 5, L=1, R=2이다.
a[L]과 a[R]을 교환한 직후의 배열을 쓰고,
각 값 7과 1을 서로 바꾸는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 52 정답과 해설

a=[2,7,1,6,5]에서 피벗은 맨 끝의 5, L=1, R=2이다.
a[L]과 a[R]을 교환한 직후의 배열을 쓰고,
각 값 7과 1을 서로 바꾸는 이유를 설명하시오.

[2,1,7,6,5]. 7은 피벗보다 커서 오른쪽, 1은 피벗 이하라서 왼쪽으로 보낸다.

해설

인덱스 1과 2의 값만 교환한다.
분할은 피벗을 기준으로 값을 보내는 단계이며, 전체 정렬이 끝난 상태는 아니다.

문제 53

mod_power(a,b,m)은 a의 b제곱을 m으로 나눈 나머지를 반환한다.
a=3, b=5, m=7이고 half=mod_power(3,2,7)=2까지 구했다.
b가 홀수이므로 결과는 (half * half * a)%m이다. 결과와 계산식을 쓰시오.
%는 나머지 연산이다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 53 정답과 해설

mod_power(a,b,m)은 a의 b제곱을 m으로 나눈 나머지를 반환한다.
a=3, b=5, m=7이고 half=mod_power(3,2,7)=2까지 구했다.
b가 홀수이므로 결과는 (half * half * a)%m이다. 결과와 계산식을 쓰시오.
%는 나머지 연산이다.

---

5. (2×2×3)%7 = 12%7 = 5.

해설

half=2를 두 번 곱하고 홀수 지수에 필요한 a=3을 한 번 곱한다.
12를 7로 나눈 나머지는 5이다.

문제 54

2×2 영상의 쿼드 트리 표현이 (0100)이다.
괄호 안은 좌상·우상·좌하·우하 순이며 0은 흰색, 1은 검은색이다.
복원된 영상의 두 행을 쓰고, 검은 칸의 (행,열)을 쓰시오. 행·열은 0부터 센다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 54 정답과 해설

2×2 영상의 쿼드 트리 표현이 (0100)이다.
괄호 안은 좌상·우상·좌하·우하 순이며 0은 흰색, 1은 검은색이다.
복원된 영상의 두 행을 쓰고, 검은 칸의 (행,열)을 쓰시오. 행·열은 0부터 센다.

첫 행 01, 둘째 행 00. 검은 칸은 (0,1).

해설

좌상 0과 우상 1이 첫 행이고 좌하 0과 우하 0이 둘째 행이다.
따라서 검은 칸은 위쪽 오른쪽, 즉 행 0과 열 1이다.

문제 55

4×4 영상의 모든 값이 0이다. 쿼드 트리 함수는 현재 영역의 모든 칸을
기준값과 비교하여 같은 색인지 확인하고, 균일하면 색 한 글자를 반환한다.
첫 호출에서 검사하는 칸 수와 반환하는 문자열을 쓰시오. 첫 칸도 검사에 포함한다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 55 정답과 해설

4×4 영상의 모든 값이 0이다. 쿼드 트리 함수는 현재 영역의 모든 칸을
기준값과 비교하여 같은 색인지 확인하고, 균일하면 색 한 글자를 반환한다.
첫 호출에서 검사하는 칸 수와 반환하는 문자열을 쓰시오. 첫 칸도 검사에 포함한다.

16칸, "0".

해설

4행×4열이므로 검사하는 칸은 16개다.
모두 0이라서 더 나누지 않고 "0" 한 글자를 반환한다.

문제 56

2×2 영상의 행이 01,10이다. 혼합 영역은 괄호로 감싸고 좌상·우상·좌하·우하 순으로 부호화하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 56 정답과 해설

2×2 영상의 행이 01,10이다. 혼합 영역은 괄호로 감싸고 좌상·우상·좌하·우하 순으로 부호화하시오.

(0110)

해설

네 칸이 각각 0,1,1,0이므로 그 순서로 적는다.

문제 57

4×4 영상의 네 행이 0011,0011,1100,1100이다. 균일하면 색 하나, 혼합이면 (좌상 우상 좌하 우하) 순으로 부호화한 결과는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 57 정답과 해설

4×4 영상의 네 행이 0011,0011,1100,1100이다. 균일하면 색 하나, 혼합이면 (좌상 우상 좌하 우하) 순으로 부호화한 결과는?

(0110)

해설

네 사분면이 각각 한 색이므로 각 사분면을 한 기호로 기록한다.

문제 58

4×4 영상 전체가 1이다. 쿼드트리 부호와 리프 수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 58 정답과 해설

4×4 영상 전체가 1이다. 쿼드트리 부호와 리프 수는?

1, 1개

해설

루트에서 균일하므로 자식을 만들지 않는다.

문제 59

2×2 영상의 각 칸까지 모두 분할했다. 내부 노드 수와 리프 수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 59 정답과 해설

2×2 영상의 각 칸까지 모두 분할했다. 내부 노드 수와 리프 수는?

1개,4개

해설

혼합 루트 하나와 네 단위 칸 리프가 있다.

문제 60

한 변 8인 영상을 끝까지 쿼드트리로 분할한다. 루트 깊이를 0이라 할 때 리프 깊이와 리프 수는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 60 정답과 해설

한 변 8인 영상을 끝까지 쿼드트리로 분할한다. 루트 깊이를 0이라 할 때 리프 깊이와 리프 수는?

깊이 3,64개

해설

변 길이 8→4→2→1이며 마지막 칸은 8^2개이다.

문제 61

$T(n)=9T(n/3)+\Theta(n)$에서 a,b,기준 지수 $\log_b a$를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 61 정답과 해설

$T(n)=9T(n/3)+\Theta(n)$에서 a,b,기준 지수 log_b a를 쓰시오.

9,3,2

해설

부분 문제 9개가 각각 n/3 크기이고 $3^2=9$이다.

문제 62

$T(n)=8T(n/2)+\Theta(n^2)$의 차수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 62 정답과 해설

$T(n)=8T(n/2)+\Theta(n^2)$의 차수를 쓰시오.

$\Theta(n^3)$

해설

기준 지수 3이 결합 지수 2보다 크므로 재귀 하위 층이 지배한다.

문제 63

$T(n)=4T(n/2)+\Theta(n^2)$의 차수를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 63 정답과 해설

$T(n)=4T(n/2)+\Theta(n^2)$의 차수를 쓰시오.

$\Theta(n^2 \log n)$

해설

각 층 비용이 $\Theta(n^2)$이고 층 수가 $\Theta(\log n)$이다.

문제 64

재귀식 $T(n)=2T(n/2)+n$, $T(1)=1$이 주어졌다. $T(2)$, $T(4)$, $T(8)$을 이 순서로 구하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 64 정답과 해설

재귀식 $T(n)=2T(n/2)+n$, $T(1)=1$이 주어졌다. $T(2)$, $T(4)$, $T(8)$을 이 순서로 구하시오.

$T(2)=4$, $T(4)=12$, $T(8)=32$

해설

$T(2)=2\times1+2=4$, $T(4)=2\times4+4=12$, $T(8)=2\times12+8=32$이다.

문제 65

병합 정렬로 [6,2,5,1]을 정렬하려 한다.
처음에 배열을 가운데에서 같은 길이로 나눌 때
왼쪽 배열과 오른쪽 배열을 각각 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 65 정답과 해설

병합 정렬로 [6,2,5,1]을 정렬하려 한다.
처음에 배열을 가운데에서 같은 길이로 나눌 때
왼쪽 배열과 오른쪽 배열을 각각 쓰시오.

왼쪽 [6,2], 오른쪽 [5,1]

해설

첫 두 원소가 왼쪽 부분이고 뒤의 두 원소가 오른쪽 부분이다.
분할 단계에서는 원소를 정렬하거나 위치를 바꾸지 않는다.

문제 66

병합 정렬에서 한 원소씩 나눈 [6], [2]를 병합하고,
[5], [1]도 병합한다. 두 병합 결과를 순서대로 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 66 정답과 해설

병합 정렬에서 한 원소씩 나눈 [6], [2]를 병합하고,
[5], [1]도 병합한다. 두 병합 결과를 순서대로 쓰시오.

[2,6], [1,5]

해설

6과 2 중 작은 값 2를 먼저 넣으면 [2,6]이 된다.
5와 1 중 작은 값 1을 먼저 넣으면 [1,5]가 된다.

문제 67

정렬된 [2,6]과 [1,5]를 오름차순으로 병합한다.
작은 값을 차례로 넣어 완성된 배열을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 67 정답과 해설

정렬된 [2,6]과 [1,5]를 오름차순으로 병합한다.
작은 값을 차례로 넣어 완성된 배열을 쓰시오.

[1,2,5,6]

해설

2와 1 중 1, 2와 5 중 2, 6과 5 중 5를 고른다.
오른쪽이 비면 왼쪽에 남은 6을 마지막에 붙인다.

문제 68

퀵 정렬의 분할이 끝나 a=[3,4,1,8,7,6], L=3, hi=5이다.
피벗 6을 제자리에 놓으려고 a[L]과 a[hi]를 교환한다.
교환 후 배열과 피벗 인덱스를 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 68 정답과 해설

퀵 정렬의 분할이 끝나 a=[3,4,1,8,7,6], L=3, hi=5이다.
피벗 6을 제자리에 놓으려고 a[L]과 a[hi]를 교환한다.
교환 후 배열과 피벗 인덱스를 쓰시오.

배열 [3,4,1,6,7,8], 피벗 인덱스 3

해설

인덱스 3의 값 8과 인덱스 5의 피벗 6을 교환한다.
피벗의 왼쪽 값은 6 이하이고 오른쪽 값은 6보다 크다.

문제 69

a=[8,4,1,3,7,6]에서 피벗은 a[5]=6, L=0, R=4이다.
L≤R이고 a[L]≤6인 동안 L을 늘린 뒤,
L≤R이고 a[R]>6인 동안 R을 줄인다. 멈춘 L, R은?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 69 정답과 해설

a=[8,4,1,3,7,6]에서 피벗은 a[5]=6, L=0, R=4이다.
L≤R이고 a[L]≤6인 동안 L을 늘린 뒤,
L≤R이고 a[R]>6인 동안 R을 줄인다. 멈춘 L, R은?

L=0, R=3

해설

a[0]=8은 6보다 크므로 L은 0에서 멈춘다.
a[4]=7은 6보다 커서 R을 3으로 줄인다. a[3]=3에서 멈춘다.

문제 70

퀵 정렬의 [0,5] 구간에서 피벗 인덱스 p=3이 확정되었다.
피벗을 제외하고 재귀 정렬할 왼쪽·오른쪽 구간을 쓰시오.
구간은 양 끝 인덱스를 포함한다.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 70 정답과 해설

퀵 정렬의 [0,5] 구간에서 피벗 인덱스 p=3이 확정되었다.
피벗을 제외하고 재귀 정렬할 왼쪽·오른쪽 구간을 쓰시오.
구간은 양 끝 인덱스를 포함한다.

왼쪽 [0,2], 오른쪽 [4,5]

해설

피벗 인덱스 3은 다시 정렬하지 않는다.
왼쪽은 [lo,p-1]=[0,2], 오른쪽은 [p+1,hi]=[4,5]이다.

문제 71

표준 병합에서 두 현재 원소 중 작은 것을 먼저 출력해도 되는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 71 정답과 해설

표준 병합에서 두 현재 원소 중 작은 것을 먼저 출력해도 되는 이유를 설명하시오.

각 입력이 정렬되어 있어 현재 원소가 그 입력의 남은 최솟값이다.

해설

두 최솟값 중 작은 값은 전체 남은 원소의 최솟값이다. 이를 반복하면 정렬 결과가 된다.

문제 72

병합의 동등 키에서 오른쪽을 우선하면 안정성이 깨지는 최소 반례를 제시하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 72 정답과 해설

병합의 동등 키에서 오른쪽을 우선하면 안정성이 깨지는 최소 반례를 제시하시오.

왼쪽 [2A], 오른쪽 [2B]이면 결과가 [2B,2A]가 된다.

해설

원래 앞에 있던 A가 뒤로 이동했다. 왼쪽을 우선하면 같은 키의 상대 순서가 유지된다.

문제 73

합병 정렬에서 반열린 구간 [lo,hi)를 쓰는 장점과 기저 조건을 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 73 정답과 해설

합병 정렬에서 반열린 구간 [lo,hi)를 쓰는 장점과 기저 조건을 설명하시오.

길이가 hi−lo이고 hi−lo≤1이면 종료한다. 분할은 [lo,mid),[mid,hi)이다.

해설

원소가 중복되거나 빠지지 않고 빈 구간도 길이 0으로 표현된다.

문제 74

최대 연속합을 왼쪽 정답과 오른쪽 정답만 비교하면 틀리는 반례를 제시하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 74 정답과 해설

최대 연속합을 왼쪽 정답과 오른쪽 정답만 비교하면 틀리는 반례를 제시하시오.

[2,3]을 둘로 나누면 각 답은 2,3이나 전체 답은 5이다.

해설

경계를 가로지르는 구간을 빼먹었다. 왼쪽 최대 접미와 오른쪽 최대 접두를 더한 후보가 필요하다.

문제 75

비어 있지 않은 최대 연속합에서 최댓값을 0으로 초기화하는 코드의 반례와 수정 원칙을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 75 정답과 해설

비어 있지 않은 최대 연속합에서 최댓값을 0으로 초기화하는 코드의 반례와 수정 원칙을 쓰시오.

[−5,−2]에서 잘못된 0이 남는다. 첫 유효 합이나 음의 무한대로 초기화한다.

해설

빈 구간을 허용하지 않는 계약을 초기값과 기저 사례에도 반영해야 한다.

문제 76

최대 교차합이 왼쪽 접미와 오른쪽 접두의 합인 이유를 연속성으로 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 76 정답과 해설

최대 교차합이 왼쪽 접미와 오른쪽 접두의 합인 이유를 연속성으로 설명하시오.

경계를 지나는 연속 구간은 반드시 mid와 mid+1을 포함한다.

해설

따라서 왼쪽 선택은 mid에서 끝나고 오른쪽 선택은 mid+1에서 시작한다. 두 쪽을 각각 최대화해 합한다.

문제 77

퀵 정렬의 탐색 조건은 L≤R and a[L]≤pivot이다.
왼쪽 조건이 거짓이면 오른쪽을 평가하지 않는다고 할 때,
a[L]을 읽기 전에 L≤R을 먼저 확인하는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 77 정답과 해설

퀵 정렬의 탐색 조건은 L≤R and a[L]≤pivot이다.
왼쪽 조건이 거짓이면 오른쪽을 평가하지 않는다고 할 때,
a[L]을 읽기 전에 L≤R을 먼저 확인하는 이유를 설명하시오.

포인터가 교차한 뒤 탐색 구간 밖의 배열 값을 읽지 않기 위해서이다.

해설

L>R이면 아직 검사할 구간이 비어 있다.
조건의 왼쪽에서 이를 확인하면 a[L] 접근을 생략하고 탐색을 끝낼 수 있다.

문제 78

빠른 거듭제곱에서 같은 절반 지수의 함수 호출을 두 번 곱하지 않고,
half=mod_power(a,b//2,m)을 한 번 호출한 뒤 half * half를 쓴다.
두 방식의 결과는 같아도 저장한 half를 재사용하는 방식이 효율적인 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 78 정답과 해설

빠른 거듭제곱에서 같은 절반 지수의 함수 호출을 두 번 곱하지 않고,
half=mod_power(a,b//2,m)을 한 번 호출한 뒤 half * half를 쓴다.
두 방식의 결과는 같아도 저장한 half를 재사용하는 방식이 효율적인 이유를 설명하시오.

같은 작은 문제를 한 번만 계산하고 저장된 결과를 두 번 사용하므로 중복 재귀 계산을 줄인다.

해설

half * half는 함수 호출 두 번이 아니라 저장된 숫자 두 개의 곱셈이다.
지수를 절반으로 줄이는 재귀는 각 단계에서 한 번만 실행된다.

문제 79

영상 배열 image[r][c]에서 r은 행, c는 열이다.
쿼드 트리의 좌하 자식은 (r+half,c,half)로 호출한다.
(c에 half를 더하는 대신) r에 half를 더하는 이유를 위치와 연결해 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 79 정답과 해설

영상 배열 image[r][c]에서 r은 행, c는 열이다.
쿼드 트리의 좌하 자식은 (r+half,c,half)로 호출한다.
(c에 half를 더하는 대신) r에 half를 더하는 이유를 위치와 연결해 설명하시오.

행 번호가 증가하면 아래로 이동하고, 좌하이므로 열은 그대로 유지해야 한다.

해설

r+half는 현재 영역의 아래 절반을 가리킨다.
c+half로 바꾸면 오른쪽으로 이동하므로 좌하가 아니라 우상이 된다.

문제 80

쿼드트리 부호화에서 균일 영역의 자식을 더 만들지 않아도 원본을 복원할 수 있는 이유는?

풀이 과정과 판단 근거를 함께 쓰시오.

문제 80 정답과 해설

쿼드트리 부호화에서 균일 영역의 자식을 더 만들지 않아도 원본을 복원할 수 있는 이유는?

영역의 위치와 크기를 알고 있으므로 색 하나로 모든 칸을 채울 수 있다.

해설

혼합 영역만 정해진 순서의 네 자식을 읽는다. 리프의 색과 재귀 위치가 함께 정보를 보존한다.

문제 81

체커보드 영상에서 쿼드트리가 압축 효과를 거의 내지 못하는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 81 정답과 해설

체커보드 영상에서 쿼드트리가 압축 효과를 거의 내지 못하는 이유를 설명하시오.

작은 영역에도 색이 섞여 결국 1×1까지 내려가야 한다.

해설

N^2개 리프 기호와 내부 구조 기호가 필요하므로 균일 영상과 달리 출력 크기가 $\Theta(N^2)$이다.

문제 82

충돌 탐색에서 같은 쿼드트리 리프의 객체끼리만 검사하는 구현의 반례를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 82 정답과 해설

충돌 탐색에서 같은 쿼드트리 리프의 객체끼리만 검사하는 구현의 반례를 설명하시오.

경계 양쪽에 있는 두 큰 객체의 영역이 경계를 넘어 겹칠 수 있다.

해설

서로 다른 리프에 등록됐더라도 경계·이웃 영역을 조회하거나 객체의 전체 경계 상자를 이용해야 한다.

문제 83

병합 정렬이 [4,1,3,2]를 [4,1]과 [3,2]로 나눈 뒤 각 절반을 정렬한다. 정렬된 두 절반과 최종 병합 결과를 쓰고, 병합 때 두 배열의 맨 앞 원소 중 어느 쪽을 먼저 고르는지 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 83 정답과 해설

병합 정렬이 [4,1,3,2]를 [4,1]과 [3,2]로 나눈 뒤 각 절반을 정렬한다. 정렬된 두 절반과 최종 병합 결과를 쓰고, 병합 때 두 배열의 맨 앞 원소 중 어느 쪽을 먼저 고르는지 설명하시오.

[1,4], [2,3] → [1,2,3,4]. 남은 두 배열의 맨 앞 원소 중 작은 값을 고른다.

해설

각 절반이 정렬되어 있으므로 남은 원소 중 최솟값은 두 맨 앞 원소 중 하나다. 둘 중 작은 것을 결과에 옮긴다.

문제 84

$T(n)=T(n-1)+n$을 직접 전개하여 차수를 증명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 84 정답과 해설

$T(n)=T(n-1)+n$을 직접 전개하여 차수를 증명하시오.

$n+(n-1)+\dots+1=\Theta(n^2)$이다.

해설

부분 문제 크기가 고정 비율 감소가 아니므로 기본 마스터 정리를 억지로 적용하지 않는다.

문제 85

정렬된 두 배열을 병합하다 한쪽 배열을 모두 사용했다. 다른 배열의 남은 원소를 순서대로 붙여도 되는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 85 정답과 해설

정렬된 두 배열을 병합하다 한쪽 배열을 모두 사용했다.
다른 배열의 남은 원소를 순서대로 붙여도 되는 이유를
설명하시오.

남은 배열도 정렬되어 있고 비교할 반대쪽 원소가 없다.

해설

병합 과정에서 앞의 작은 값들을 이미 결과에 넣었다.
남은 배열 내부의 순서를 유지하면 전체 오름차순도 유지된다.

문제 86

[1,5]와 [2,4]는 각각 정렬되어 있다. 그대로 이어 붙이면 [1,5,2,4]가 된다. 이 결과가 정렬되지 않은 이유와 올바르게 병합하는 방법을 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 86 정답과 해설

[1,5]와 [2,4]는 각각 정렬되어 있다. 그대로 이어 붙이면
[1,5,2,4]가 된다. 이 결과가 정렬되지 않은 이유와
올바르게 병합하는 방법을 설명하시오.

5 뒤의 2가 더 작다. 양쪽 현재 값 중 작은 값을 먼저 넣는다.

해설

각 부분이 정렬되어 있어도 두 부분의 값 범위는 겹칠 수 있다.
양쪽 앞의 값을 비교해 1, 2, 4, 5 순서로 결과를 만든다.

문제 87

(점수,이름)이 [(2,A),(1,B),(2,C)]인 배열을 점수로 정렬한다.
안정 정렬은 같은 점수의 원래 순서를 유지한다.
안정적인 병합 정렬의 결과와 A·C의 순서를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 87 정답과 해설

(점수,이름)이 [(2,A),(1,B),(2,C)]인 배열을 점수로 정렬한다.
안정 정렬은 같은 점수의 원래 순서를 유지한다.
안정적인 병합 정렬의 결과와 A·C의 순서를 설명하시오.

[(1,B),(2,A),(2,C)]. 같은 점수에서 A가 C보다 앞선다.

해설

점수 1인 B를 먼저 놓고 점수 2인 A와 C를 놓는다.
A가 원래 먼저였으므로 같은 점수에서도 A, C 순서를 보존한다.

문제 88

양 끝 인덱스를 포함하는 quick_sort(a,lo,hi)에서 lo≥hi이면 바로 종료한다. lo=hi일 때와 lo>hi일 때 각각 종료해도 되는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 88 정답과 해설

양 끝 인덱스를 포함하는 quick_sort(a,lo,hi)에서 lo≥hi이면 바로 종료한다. lo=hi일 때와 lo>hi일 때 각각 종료해도 되는 이유를 설명하시오.

lo=hi는 원소 1개, lo>hi는 빈 구간이므로 이미 정렬되었다.

해설

원소가 하나이면 비교할 다른 값이 없고, 빈 구간에는 원소가 없다. 두 경우 모두 더 나누거나 피벗을 고를 필요가 없다.

문제 89

퀵 정렬에서 피벗이 인덱스 p에 확정되면
[lo,p-1]과 [p+1,hi]만 재귀 정렬한다.
피벗을 다시 포함하지 않는 이유를 설명하시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 89 정답과 해설

퀵 정렬에서 피벗이 인덱스 p에 확정되면
[lo,p-1]과 [p+1,hi]만 재귀 정렬한다.
피벗을 다시 포함하지 않는 이유를 설명하시오.

피벗은 이미 제자리이고, 남은 두 구간만 정렬하면 된다.

해설

분할 후 피벗 왼쪽은 피벗 이하, 오른쪽은 피벗보다 크다.
확정된 피벗을 제외하면 재귀 구간이 작아지고 종료할 수 있다.

문제 90

분할 정복 함수의 기저 사례를 n==1로만 둘 때 빈 입력에서 생길 수 있는 문제와 수정안을 쓰시오.

풀이 과정과 판단 근거를 함께 쓰시오.

문제 90 정답과 해설

분할 정복 함수의 기저 사례를 n==1로만 둘 때 빈 입력에서 생길 수 있는 문제와 수정안을 쓰시오.

n=0을 계속 둘로 나눠도 0이어서 종료하지 않을 수 있다. n≤1을 기저로 처리한다.

해설

빈 입력의 반환값은 문제마다 다르다. 정렬은 빈 배열을 반환하지만 비어 있지 않은 최대 연속합은 빈 입력을 금지할 수 있다.

문제 91

오름차순 정수 배열 a,b를 합친 정렬 결과를 merge_arrays로 구한다. 각 길이는 0..500. 같은 값이면 a를 먼저 쓴다. C는 충분한 out 배열에 기록한다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 91 정답과 해설

정렬 배열 병합

두 배열의 맨 앞 미처리 원소를 비교하고 작은 쪽을 출력한다.

해설

시간 O(n+m).

문제 91 Python 구현

정렬 배열 병합

```python
def merge_arrays(a, b):
    i = j = 0
    out = []
    while i < len(a) and j < len(b):
        if a[i] <= b[j]:
            out.append(a[i])
            i += 1
        else:
            out.append(b[j])
            j += 1
    while i < len(a):
        out.append(a[i])
        i += 1
    while j < len(b):
        out.append(b[j])
        j += 1
    return out
```

시간 O(n+m).

문제 91 C 구현

정렬 배열 병합

```c
void merge_arrays(const int a[], int n, const int b[],
                  int m, int out[]) {
    int i=0, j=0, k=0;
    while (i<n && j<m) {
        if (a[i]<=b[j]) out[k++]=a[i++];
        else out[k++]=b[j++];
    }
    while (i<n) out[k++]=a[i++];
    while (j<m) out[k++]=b[j++];
}
```

시간 O(n+m).

문제 92

길이 0..1000의 정수 배열을 병합 정렬하시오.
Python merge_sort(a)는 새 배열을 반환한다.
C merge_sort(a,lo,hi,tmp)는 lo 이상 hi 미만인 [lo,hi)를 제자리 정렬한다.
전체 배열의 C 호출은 merge_sort(a,0,n,tmp)이며 tmp는 길이 n이다.
Python의 merge_arrays는 제공된다. Python 또는 C로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 92 정답과 해설

병합 정렬

길이 1 이하에서 멈추고 양쪽 정렬 결과를 안정적으로 병합한다.

해설

시간 O(n log(n+1)). Python은 앞 문항 merge_arrays를 사용한다.

문제 92 Python 구현

병합 정렬

```python
def merge_sort(a):
    if len(a) <= 1:
        return a[:]
    mid = len(a) // 2
    return merge_arrays(merge_sort(a[:mid]),
                        merge_sort(a[mid:]))
```

시간 O(n log(n+1)). Python은 앞 문항 merge_arrays를 사용한다.

문제 92 C 구현

병합 정렬

```c
void merge_sort(int a[], int lo, int hi, int tmp[]) {
    if (hi-lo <= 1) return;
    int mid=(lo+hi)/2;
    merge_sort(a,lo,mid,tmp);
    merge_sort(a,mid,hi,tmp);
    int i=lo, j=mid, k=lo;
    while (i<mid && j<hi) {
        if (a[i]<=a[j]) tmp[k++]=a[i++];
        else tmp[k++]=a[j++];
    }
    while (i<mid) tmp[k++]=a[i++];
    while (j<hi) tmp[k++]=a[j++];
    for (i=lo; i<hi; ++i) a[i]=tmp[i];
}
```

시간 O(n log(n+1)). Python은 앞 문항 merge_arrays를 사용한다.

문제 93

정수 배열의 [lo,hi)에서 lo<mid<hi다. 양쪽에 적어도 한 원소를 포함하는 최대 연속합 cross_sum을 작성한다. n≤1000, 값은 $-10^6..10^6$. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 93 정답과 해설

경계를 가로지르는 합

왼쪽의 최대 접미사 합과 오른쪽의 최대 접두사 합을 더한다.

해설

시간 O(hi−lo). 모두 음수여도 빈 구간을 허용하지 않는다.

문제 93 Python 구현

경계를 가로지르는 합

```python
def cross_sum(a, lo, mid, hi):
    total = 0
    left = a[mid - 1]
    for i in range(mid - 1, lo - 1, -1):
        total += a[i]
        left = max(left, total)
    total = 0
    right = a[mid]
    for i in range(mid, hi):
        total += a[i]
        right = max(right, total)
    return left + right
```

시간 O(hi−lo). 모두 음수여도 빈 구간을 허용하지 않는다.

문제 93 C 구현

경계를 가로지르는 합

```c
long long cross_sum(const int a[], int lo, int mid, int hi) {
    long long sum=0, left=a[mid-1], right=a[mid];
    for (int i=mid-1; i>=lo; --i) {
        sum+=a[i];
        if (sum>left) left=sum;
    }
    sum=0;
    for (int i=mid; i<hi; ++i) {
        sum+=a[i];
        if (sum>right) right=sum;
    }
    return left+right;
}
```

시간 O(hi−lo). 모두 음수여도 빈 구간을 허용하지 않는다.

문제 94

비어 있지 않은 배열 [lo,hi)의 최대 연속합 max_subarray를 분할 정복으로 구하시오. 빈 구간은 금지한다. $n \le 1000$, 값은 $-10^6..10^6$. cross_sum은 제공된다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 94 정답과 해설

분할 정복 최대 연속합

왼쪽, 오른쪽, 경계를 가로지르는 세 후보 중 최댓값을 반환한다.

해설

시간 O(n log n).

문제 94 Python 구현

분할 정복 최대 연속합

```python
def max_subarray(a, lo, hi):
    if hi - lo == 1:
        return a[lo]
    mid = (lo + hi) // 2
    return max(max_subarray(a, lo, mid),
               max_subarray(a, mid, hi),
               cross_sum(a, lo, mid, hi))
```

시간 O(n log n).

문제 94 C 구현

분할 정복 최대 연속합

```c
long long max_subarray(const int a[], int lo, int hi) {
    if (hi-lo == 1) return a[lo];
    int mid=(lo+hi)/2;
    long long left=max_subarray(a,lo,mid);
    long long right=max_subarray(a,mid,hi);
    long long cross=cross_sum(a,lo,mid,hi);
    long long best=left>right ? left:right;
    return best>cross ? best:cross;
}
```

시간 O(n log n).

문제 95

a의 b제곱을 m으로 나눈 나머지를 반환하는 mod_power(a,b,m)을 작성하시오.
$0 \le a \le 10^9$, $0 \le b \le 10^9$, $1 \le m \le 10^9$이다. b=0이면 1%m을 반환한다.
그 외에는 b//2의 재귀 결과를 한 번만 구해 half에 저장하고 제곱한다.
b가 홀수이면 a를 한 번 더 곱하며, 곱셈 후마다 m으로 나눈 나머지를 저장한다.
Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 95 정답과 해설

빠른 거듭제곱과 나머지

절반 지수의 결과를 한 번 구해 저장하고, 제곱한 뒤 홀수일 때만 a를 추가로 곱한다.

해설

시간 O(log(b+1)). 지수를 절반으로 줄이는 호출은 한 번이며, 저장한 half를 재사용한다.

문제 95 Python 구현

빠른 거듭제곱과 나머지

```python
def mod_power(a, b, m):
    if b == 0:
        return 1 % m
    half = mod_power(a, b // 2, m)
    result = (half * half) % m
    if b % 2 == 1:
        result = (result * a) % m
    return result
```

시간 O(log(b+1)). 지수를 절반으로 줄이는 호출은 한 번이며, 저장한 half를 재사용한다.

문제 95 C 구현

빠른 거듭제곱과 나머지

```c
int mod_power(int a, int b, int m) {
    if (b == 0) return 1 % m;
    long long half = mod_power(a, b / 2, m);
    long long result = (half * half) % m;
    if (b % 2 == 1) result = (result * a) % m;
    return (int)result;
}
```

시간 O(log(b+1)). 지수를 절반으로 줄이는 호출은 한 번이며, 저장한 half를 재사용한다.

문제 96

$n=2^k$, $1 \le n \le 32$인 0/1 정사각형을 부호화한다. 균일하면 값 하나, 아니면 (왼위 오른위 왼아래 오른아래) 순서로 붙인다. quad(a,r,c,n)을 작성한다. C a의 열 폭은 32다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 96 정답과 해설

쿼드트리 부호화

균일성을 검사하고 서로 다르면 네 사분면을 재귀 호출한다.

해설

이 전체 균일성 검사 구현은 $O(n^2 \log(n+1))$ 시간 상한. .

문제 96 Python 구현

쿼드트리 부호화

```python
def quad(a, r, c, n):
    value = a[r][c]
    if all(a[i][j] == value
           for i in range(r, r + n)
           for j in range(c, c + n)):
        return str(value)
    h = n // 2
    return ('(' + quad(a,r,c,h) + quad(a,r,c+h,h)
            + quad(a,r+h,c,h) + quad(a,r+h,c+h,h) + ')')
```

이 전체 균일성 검사 구현은 $O(n^2 \log(n+1))$ 시간 상한. .

문제 96 C 구현

쿼드트리 부호화

```c
/* out has >= 4*n*n+1 bytes; start *p=0. */
void quad(const int a[][32], int r, int c, int n,
          char out[], int *p) {
    int same=1, value=a[r][c];
    for (int i=r; i<r+n; ++i)
        for (int j=c; j<c+n; ++j)
            if (a[i][j]!=value) same=0;
    if (same) out[(*p)++]=(char)('0'+value);
    else {
        int h=n/2; out[(*p)++]='(';
        quad(a,r,c,h,out,p); quad(a,r,c+h,h,out,p);
        quad(a,r+h,c,h,out,p); quad(a,r+h,c+h,h,out,p);
        out[(*p)++]=')';
    }
    out[*p]='\0';
}
```

이 전체 균일성 검사 구현은 $O(n^2 \log(n+1))$ 시간 상한. .

문제 97

정수 배열 a의 [lo,hi]를 다음 순서대로 분할하는 partition(a,lo,hi)를 작성하시오.
양 끝을 포함하며 0≤lo≤hi<n≤1000이다.
① 가운데 인덱스 (lo+hi)//2의 값을 hi로 교환해 피벗으로 둔다.
L=lo, R=hi-1이다.
② L≤R이고 a[L]≤피벗인 동안 L을 계속 증가시킨다.
이어 L≤R이고 a[R]>피벗인 동안 R을 계속 감소시킨다.
③ L>R이면 탐색을 끝내고, 아니면 a[L]과 a[R]을 교환한 뒤 ②를 반복한다.
④ a[L]과 a[hi]를 교환하고 L을 반환한다. 구간 밖의 값은 바꾸지 않는다.
Python 또는 C로 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

예: partition([8,4,6,3,7,1],0,5) → 3
분할 후 a = [3,4,1,6,7,8]

문제 97 정답과 해설

가운데 값 6을 끝으로 옮긴다: [8,4,1,3,7,6]
L=0의 8과 R=3의 3을 교환한다: [3,4,1,8,7,6]
L=3, R=2로 교차하면 8과 피벗 6을 교환한다.
반환값은 3, 분할 후 배열은 [3,4,1,6,7,8]이다.
원소가 하나이면 그 인덱스를 반환하고 배열은 그대로다.

문제 97 Python 구현

퀵 정렬의 분할 함수

```python
def partition(a, lo, hi):
    m = (lo + hi) // 2
    a[m], a[hi] = a[hi], a[m]
    pivot = a[hi]
    L, R = lo, hi - 1
    while True:
        while L <= R and a[L] <= pivot:
            L += 1
        while L <= R and a[R] > pivot:
            R -= 1
        if L > R:
            break
        a[L], a[R] = a[R], a[L]
    a[L], a[hi] = a[hi], a[L]
    return L
```

길이가 k인 구간에서 시간 O(k). 같은 값은 피벗의 왼쪽으로 보낸다.

문제 97 C 구현

퀵 정렬의 분할 함수

```c
int partition(int a[], int lo, int hi) {
    int m = lo + (hi - lo) / 2;
    int t = a[m]; a[m] = a[hi]; a[hi] = t;
    int pivot = a[hi], L = lo, R = hi - 1;
    while (1) {
        while (L <= R && a[L] <= pivot) L++;
        while (L <= R && a[R] > pivot) R--;
        if (L > R) break;
        t = a[L]; a[L] = a[R]; a[R] = t;
    }
    t = a[L]; a[L] = a[hi]; a[hi] = t;
    return L;
}
```

길이가 k인 구간에서 시간 O(k). 같은 값은 피벗의 왼쪽으로 보낸다.

문제 98

정수 배열 a의 [lo,hi]를 제자리 오름차순으로 정렬하는 quick_sort(a,lo,hi)를 작성하시오.
양 끝을 포함하며 길이 n은 0..1000이다.
전체 배열의 첫 호출은 (a,0,n-1)이다. 빈 배열은 lo=0, hi=-1로 호출하며 즉시 끝낸다.
앞 문항의 partition이 Python과 C 실행 환경에 제공된다.
lo≥hi이면 종료하고, 피벗 인덱스 p를 제외한
[lo,p-1]과 [p+1,hi]를 재귀 정렬한다.
구간 밖의 값은 바꾸지 않는다. 반환값 대신 배열 a를 검사한다.
sort·sorted·qsort 등 정렬 라이브러리는 쓰지 않는다.
Python 또는 C로 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

예: quick_sort([8,4,6,3,7,1],0,5)
정렬 후 a = [1,3,4,6,7,8]

문제 98 정답과 해설

분할 후 피벗 6의 자리는 p=3으로 확정된다.
왼쪽 [3,4,1]과 오른쪽 [7,8]을 각각 정렬한다.
호출 구간은 [lo,p-1]과 [p+1,hi]이다.
빈 구간이나 원소 한 개이면 즉시 종료한다.
평균 O(n log n), 최악 $O(n^2)$이다.
동작 검사는 결과를 확인한다. 지정 알고리즘은 풀이로도 점검한다.

문제 98 Python 구현

퀵 정렬

```python
def quick_sort(a, lo, hi):
    if lo >= hi:
        return
    p = partition(a, lo, hi)
    quick_sort(a, lo, p - 1)
    quick_sort(a, p + 1, hi)
```

평균 O(n log n), 최악 O(n²). 가운데 인덱스의 값이 중앙값이라는 보장은 없다.

문제 98 C 구현

퀵 정렬

```c
void quick_sort(int a[], int lo, int hi) {
    if (lo >= hi) return;
    int p = partition(a, lo, hi);
    quick_sort(a, lo, p - 1);
    quick_sort(a, p + 1, hi);
}
```

평균 O(n log n), 최악 O(n²). 가운데 인덱스의 값이 중앙값이라는 보장은 없다.

문제 99

0/1 정사각형을 균일할 때까지 네 등분한 뒤 값 1인 최종 조각의 수를 반환한다. $n=2^k \le 32$. white_count(a,r,c,n)을 작성한다. C 열 폭은 32다. Python 또는 C 중 하나로 함수를 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 99 정답과 해설

단색 종이 조각 수

균일한 1 영역은 한 조각, 균일한 0 영역은 0개로 세고 나머지는 네 결과를 합한다.

해설

시간 $O(n^2 \log(n+1))$ 상한. 1인 칸 수와 조각 수는 다르다.

문제 99 Python 구현

단색 종이 조각 수

```python
def white_count(a, r, c, n):
    value = a[r][c]
    if all(a[i][j] == value
           for i in range(r, r+n) for j in range(c, c+n)):
        return value
    h = n // 2
    return (white_count(a,r,c,h) + white_count(a,r,c+h,h)
            + white_count(a,r+h,c,h)
            + white_count(a,r+h,c+h,h))
```

시간 $O(n^2 \log(n+1))$ 상한. 1인 칸 수와 조각 수는 다르다.

문제 99 C 구현

단색 종이 조각 수

```c
int white_count(const int a[][32], int r, int c, int n) {
    int same=1, value=a[r][c];
    for (int i=r; i<r+n; ++i)
        for (int j=c; j<c+n; ++j)
            if (a[i][j]!=value) same=0;
    if (same) return value;
    int h=n/2;
    return white_count(a,r,c,h)+white_count(a,r,c+h,h)
         + white_count(a,r+h,c,h)+white_count(a,r+h,c+h,h);
}
```

시간 $O(n^2 \log(n+1))$ 상한. 1인 칸 수와 조각 수는 다르다.

문제 100

역전 쌍은 원래 배열에서 앞의 수가 뒤의 수보다 큰 두 원소의 쌍입니다.
원래 배열의 왼쪽·오른쪽 부분을 각각 오름차순으로 정렬한 a, b가 주어집니다.
a에서 하나, b에서 하나를 골라 a[i] > b[j]인 쌍만 셉니다. 같은 값은 세지 않습니다.

예: a=[1,4,7], b=[2,3,8]이면
(4,2), (4,3), (7,2), (7,3)의 4쌍입니다.

cross_inversions(a,b)를 병합처럼 순회하여 작성하시오.
각 길이는 0~1000이며, 한쪽이 비면 0을 반환합니다. Python 또는 C로 작성하시오.

함수의 동작, 경계 처리, 시간 복잡도를 함께 작성하시오.

문제 100 정답과 해설

병합 중 역전 쌍 세기

오른쪽 값이 더 작으면 남은 왼쪽 원소 모두와 역전 쌍을 만든다.

해설

a=[1,4,7], b=[2,3,8]에서 i=j=count=0으로 시작합니다.
1 <= 2: i만 늘려 4를 봅니다. 아직 역전 쌍은 없습니다.
4 > 2: 남은 왼쪽 [4,7] 모두 2보다 큽니다. 2쌍을 더하고 j를 늘립니다.
4 > 3: 다시 [4,7] 모두 3보다 큽니다. 2쌍을 더하고 j를 늘립니다.
4 <= 8, 7 <= 8: i만 늘리다가 왼쪽이 끝납니다. 총 4쌍입니다.

왼쪽 길이를 n, 오른쪽 길이를 m이라 하면 더할 개수는 n-i개입니다.
각 반복에서 i 또는 j가 한 칸 전진하므로 시간은 O(n+m)입니다.

문제 100 Python 구현

병합 중 역전 쌍 세기

```python
def cross_inversions(a, b):
    i = j = count = 0
    while i < len(a) and j < len(b):
        if a[i] <= b[j]:
            i += 1
        else:
            count += len(a) - i
            j += 1
    return count
```

시간 O(n+m). 같은 값은 역전 쌍으로 세지 않는다.

문제 100 C 구현

병합 중 역전 쌍 세기

```c
long long cross_inversions(const int a[], int n,
                           const int b[], int m) {
    int i=0, j=0;
    long long count=0;
    while (i<n && j<m) {
        if (a[i]<=b[j]) ++i;
        else { count+=n-i; ++j; }
    }
    return count;
}
```

시간 O(n+m). 같은 값은 역전 쌍으로 세지 않는다.