

자료구조론
국가 전산 7급
2008년 7월 26일

☞ 채용인원/합격선(17명/73.57점) - 양성 72.85점 / 장애 66.42점 ☞

1. C 언어를 사용하여 100행 50열로 이루어진 행렬을 2차원 배열 A[100][50]에 저장하려고 한다. 10행 5열에 정수 값 78을 저장하는 방법으로 옳지 않은 것은? [2008년 전산 7급]

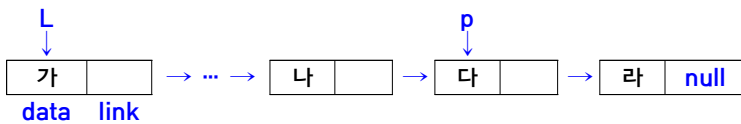
- ① A[9][4] = 78; ② *(A[0] + 454) = 78;
- ③ *(&A[9][0] + 4) = 78; ④ *(&A[0][0] + 904) = 78;

☞ A[100][50] → A[9][4] 이전까지의 배열 요소 수 = 50 * 9 + 4 = 454

A[0][0]	A[0][1]	A[0][2]	A[0][3]	A[0][4]	A[0][49]
A[1][0]	A[1][1]	A[1][2]	A[1][3]	A[1][4]	A[1][49]
A[2][0]	A[2][1]	A[2][2]	A[2][3]	A[2][4]	A[2][49]
:	:	:	:	:	:
A[9][0]	A[9][1]	A[9][2]	A[9][3]	A[9][4]	A[9][49]
:	:	:	:	:	:
A[99][0]	A[99][1]	A[99][2]	A[99][3]	A[99][4]	A[99][49]

정답 : ④

2. 다음과 같은 단순 연결리스트에 대해, 아래와 같은 C 언어로 작성된 프로그램을 수행한 후 포인터 temp가 가리키는 노드는? [2008년 전산 7급]



```
for(temp = L; temp->link != p; temp = temp->link) ;
temp->link = p->link;
```

- ① 가 ② 나
- ③ 다 ④ 라

☞ 포인터 p가 가리키는 노드(다)를 삭제하는 알고리즘이다.

• temp가 노드 '나'를 가리킬 때 조건 temp->link == p는 거짓, 반복문을 벗어난다.

정답 : ②

3. N개의 원소를 가지는 배열을 합병정렬 알고리즘으로 정렬하고자 한다. 이 알고리즘에서 값을 비교하는 횟수를 C(N)이라고 할 때, 순환식으로 표현된 C(N)과 시간복잡도는?(단, $N=2^k$ 이고, $C(1)=0$ 이라고 가정한다) [2008년 전산 7급]

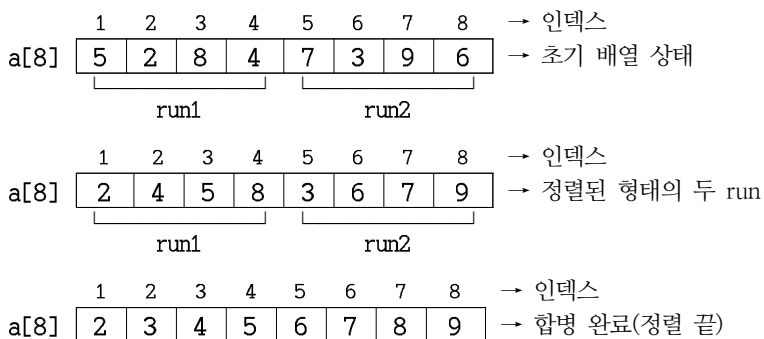
- | | |
|----------------------|---------------|
| C(N) | 시간복잡도 |
| ① $C(N)=C(N/2)+N-1$ | $O(\log N)$ |
| ② $C(N)=C(N/2)+1$ | $O(\log N)$ |
| ③ $C(N)=2C(N/2)+N-1$ | $O(N \log N)$ |
| ④ $C(N)=2C(N/2)+1$ | $O(N \log N)$ |

☞ 순환합병정렬 - 어려운 문제

//순환합병정렬은 분할과 정복의 원리로 진행된다.

- 분할 : 배열을 2개의 부분 배열(run)로 분할한다.
- 정복 : 각 run을 정렬한다. → 순환 호출 실시
- 합병 : 각 run을 하나의 정렬된 리스트로 합병한다.

[예] N = 8인 경우 순환합병정렬 과정



∴ 비교횟수 = run1 정렬시간 + run2 정렬시간 + 두 run 합병시간

$$C(N) = C(N/2) + C(N/2) + (N - 1) = 2C(N/2) + N - 1$$

◆ 추가 설명

- ㉠ 각 run의 정렬 시간이 $C(N/2)$: N개의 원소 정렬 시간이 $C(N)$ 이므로
- ㉡ 두 run의 합병 시간이 $(N - 1)$: N개의 원소에서 마지막 1개는 비교되지 않으므로

◆ 합병정렬의 시간복잡도는 $O(N \log N)$ 라는 것은 잘 알고 있는 부분이다.

[tip 1] 문제의 조건에서 $N=2^k$ 은 배열의 원소 수가 2의 거듭제곱임을 나타낸다.

[tip 2] 문제에서 $N=2^k$ 이 아니면, $C(N) = C(\lfloor N/2 \rfloor) + C(\lceil N/2 \rceil) + (N - 1) \rightarrow$ 어려운 부분!

4. 다음은 C 언어로 작성된 프로그램의 일부이다. 이 프로그램의 시간복잡도는? [2008년 전산 7급]

```

for(i = 1; i < n; i++)
{
    m = 1;
    while(m<i) m = m * 2;
}
    
```

- ① $O(n^2)$ ② $O(\log n)$
- ③ $O(n \log n)$ ④ $O(n)$

♣ $\log_2 n$ 이란?

단계수 k	2^0	2^1	2^2	2^3	2^4	2^5	...	2^k
2배씩 증가	1	2	4	8	16	32	...	n

$2^k = n$

$k = \log_2 n \rightarrow$ 어떤 값이 2배씩 증가하여 n이 되면, 단계수 k는 $\log_2 n$ 이 된다.

```

for(i = 1; i < n; i++)           //바깥 루프 : 단계수 n
{
    m = 1;
    while(m<i) m = m * 2;       //안쪽 루프 : 단계수  $\log_2 n \rightarrow$  2배씩 증가하므로
}
    
```

\therefore 시간복잡도 : $O(n \log_2 n)$



주어진 문제에서는 시간복잡도 분석에서 로그함수의 밑 2가 생략되어 있을 뿐이다.

5. 다음과 같은 7개의 10진수를 입력받아 3단계의 기수정렬을 수행하여 오름차순으로 정렬하려고 한다. 두 번째 단계를 수행한 후 부분 정렬되어진 결과는? [2008년 전산 7급]

379 - 508 - 706 - 193 - 984 - 155 - 133

- ① 508 - 706 - 155 - 133 - 379 - 984 - 193
- ② 706 - 508 - 155 - 133 - 379 - 193 - 984
- ③ 508 - 706 - 133 - 155 - 379 - 193 - 984
- ④ 706 - 508 - 133 - 155 - 379 - 984 - 193

☞ 기수정렬

입력리스트	Pass 1큐	1차 재배열	Pass 2큐	2차 재배열
	0 :		0 : 706 508	
379	1 :	193	1 :	706
508	2 :	133	2 :	508
706	3 : 193 133	984	3 : 133	133
193	4 : 984	155	4 :	155
984	5 : 155	706	5 : 155	379
155	6 : 706	508	6 :	984
133	7 :	379	7 : 379	193
	8 : 508		8 : 984	
	9 : 379		9 : 193	

↓
일의 자리값을
기준으로 분배

↓
십의 자리값을
기준으로 분배

↓
두 번째
단계 결과

// 기수정렬 정리

최대유효숫자(Most Significant Digit; MSD) 정렬	최대 유효키를 먼저 적용한다.
최소유효숫자(Least Significant Digit; LSD) 정렬	최소 유효키를 먼저 적용한다.

- 10진수 321에서 일의 자리인 1이 **최소** 유효키에 해당하고
- 10진수 321에서 백의 자리인 3이 **최대** 유효키에 해당한다.
- LSD 정렬이 MSD 정렬에 비해 간단하다. 즉, 오버헤드가 적다.