자료구조론	국가 전산 7급	2010년 7월 24일

♤ 채용인원/합격선(13명/76.42점) - 양성 75.0점 / 장애 63.57점 ♤

1. 크기가 11인 해시테이블이 있고, 해시함수로 h(k) = k mod 11을 사용한다. 여기서 mod는 모듈로(modulo) 함수를 의미한다. 하나의 해시 값에 대해 두 개씩의 슬롯이 할당되어 있고, 오버플로가 발생하면 다음 빈 슬롯에 저장하는 선형조사법(linear probe)을 사용한다고 하자. 데이터가 다음과 같은 순서로 입력된다고 할 때, 원래 계산된 슬롯에 저장되지 않는 데이터의 개수는? [2010년 전산 7급]

54, 27, 70, 55, 13, 2, 37, 23, 33, 44, 45, 77, 56, 6, 9

① 3 ② 4 ③ 5 ④ 6

♤ 해시테이블 – 각 버킷은 2개의 슬롯으로 구성

• 해시함수 : h(k) = k mod 11, 오버플로 발생시 선형조사법 적용

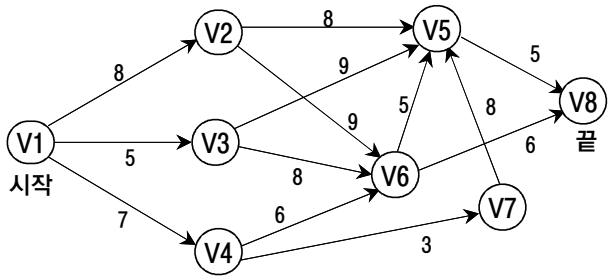
주소	0	1	2	3	4	5	6	7	8	9	10
슬롯 1	55	23	13	**45**	70	27	6			9	54
슬롯 2	33	**44**	2	**77**	37	**56**					

h(54) = 54 mod 11 = 10
h(27) = 27 mod 11 = 5
h(70) = 70 mod 11 = 4
h(55) = 55 mod 11 = 0
h(13) = 13 mod 11 = 2
h(2) = 2 mod 11 = 2
h(37) = 37 mod 11 = 4
h(23) = 23 mod 11 = 1
h(33) = 33 mod 11 = 0
h(44) = 44 mod 11 = 0 → **오버플로 발생, 다른 버킷주소에 저장됨**
h(45) = 45 mod 11 = 1 → **오버플로 발생, 다른 버킷주소에 저장됨**
h(77) = 77 mod 11 = 0 → **오버플로 발생, 다른 버킷주소에 저장됨**
h(56) = 56 mod 11 = 1 → **오버플로 발생, 다른 버킷주소에 저장됨**
h(6) = 6 mod 11 = 6
h(9) = 9 mod 11 = 9

• 원래 계산된 슬롯에 저장되지 않는 데이터 : 44, 45, 77, 56 – **4개**

정답 : ②

2. 다음 AOE(activity on edge) 네트워크로 표현된 작업들은 병렬로 수행된다. 이 그래프에서 나타나는 임계경로(critical path)는? [2010년 전산 7급]

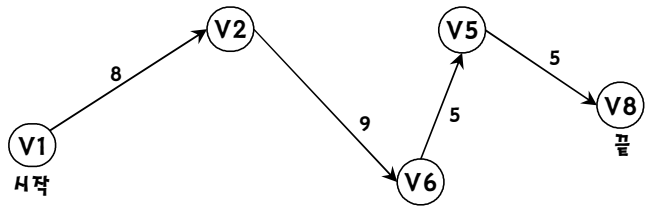


① V1 - V3 - V6 - V5 - V8
② V1 - V4 - V6 - V5 - V8
③ V1 - V2 - V6 - V5 - V8
④ V1 - V4 - V7 - V5 - V8

♤ AOE 네트워크에 대한 임계경로

임계경로 (critical path)	**임계경로**는 프로젝트를 종료하는데 필요한 **최소시간**으로 시작정점에서 종료정점까지 **최장경로길이**가 된다.

• 임계경로 : V1 - V2 - V6 - V5 - V8



[비임계경로가 제거된 그래프]

• 임계경로상의 작업이 지연되면 전체 프로젝트를 완료하는 시간은 그 만큼 길어진다.

정답 : ③

3. 다음 코드의 시간복잡도를 바르게 나타낸 것은? [2010년 전산 7급]

```
for(i = 1; i <= n; i++)
    for(j = 1; j <= n; j = j + i)
        for(k = 1; k <= n; k++)
            x = x + k + 1;
```

① $\Theta(n^2)$ ② $\Theta(n^2 \log n)$

③ $\Theta(n^3)$ ④ $\Theta(n^3 \log n)$

♤ **시간복잡도 – 매우 어려운 문제라고 생각될 수 있다.**

• 먼저, 주어진 코드에 대한 시간복잡도는 어려운 문제일 수 있다.

• **두 번째 for문**의 매개변수 j의 증가분 값에 따른 for문의 **평균 반복횟수**를 구하는 것이 관건이다.

i	j의 증가(j = j + i)	두 번째 for문의 반복횟수
1	1 2 3 4 5 6 7 ..... n	n
2	1 3 5 7 9 ..... n	n/2
3	1 4 7 10.....n	n/3
:		
n	1	1

• **두 번째 for문의 평균 반복횟수** = (n + n/2 + n/3 + ... + 1) / n

= n(1 + 1/2 + 1/3 + ... + 1/n) / n

= 1 + 1/2 + 1/3 + ... + 1/n

• 여기서, 1 + 1/2 + 1/3 + ... + 1/n = **Θ(log n)** → 별도로 정리한다.

// 주어진 코드의 시간복잡도

```
for(i = 1; i <= n; i++)          → Θ(n)
    for(j = 1; j <= n; j = j + I)  → Θ(log n)
        for(k = 1; k <= n; k++)    → Θ(n)
            x = x + k + 1;
```

∴ **복잡도** = $\Theta(n^2 \log n)$

◈ 추가 설명

• 주어진 문제에서 log의 밑수(base)가 얼마인지 제시되지 않았다.
• log의 밑수를 2라고 가정하면 이 문제는 쉽게 증명할 수가 있다.
• 단지, 전산 시험에서는 log의 밑수를 2라고 가정하고 푼다.

정답 : ②

◈ y = 1 + 1/2 + 1/3 + ... + 1/n

• 먼저, 수식의 합인 y값은 일반항으로 나타낼 수 없다.
• 수식의 합인 y값은 근사값으로 표현할 수 있다.
• 만약, n이 작은 값이 아니면

---

y = 1 + 1/2 + 1/3 + ... + 1/n ≈ ln n

→ 여기서, 'ln n'은 자연대수를 나타낸다.($\ln n = \log_e n$)

ln n = ln 10 ≈ 2.3025851

• 위의 개념은 자료구조에서 알고리즘을 분석할 때 사용되는 것이다.

◈ 기호 ≈

• 기호 ≈는 "근사적으로 같다"라는 의미이다.
• 기호 ≈는 어떤 값을 근사값으로만 표현할 수 있을 때 사용한다.

π ≈ 3.141592.....

4. 아래 코드는 가용공간 리스트로부터 노드를 할당받아 이중연결리스트에 새로운 값 x를 삽입하기 위한 함수의 일부이다. 할당받은 노드에 대한 포인터는 new이고, 이중연결리스트에서의 삽입 위치는 pre가 가리키는 노드 뒤라고 하자. 밑줄 친 ㉠, ㉡에 알맞은 문장은? [2010년 전산 7급]

```
new ← getNode();
new.data ← x;
[          ㉠          ]
[          ㉡          ]
pre.rlink ← new;
new.llink ← pre;
```

① ㉠ new.rlink ← pre.rlink;
㉡ new.rlink.llink ← new.llink;

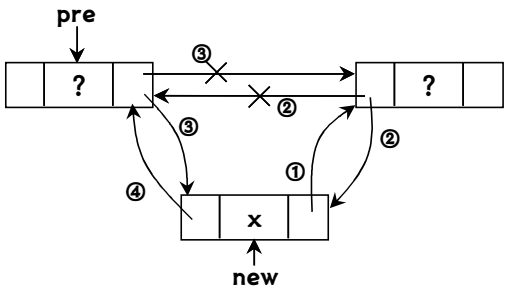
② ㉠ new.llink ← pre.rlink;
㉡ new.llink.rlink ← new;

③ ㉠ new.llink ← pre.rlink;
㉡ new.llink.rlink ← new.rlink;

④ ㉠ new.rlink ← pre.rlink;
㉡ new.rlink.llink ← new;

♤ **이중 연결리스트에서 노드 삽입 – pre가 가리키는 노드 다음에 새로운 new 노드 삽입**

• 다음 그림에서 지정한 번호는 실행되는 순서이다. ×는 연결이 없어지는 것을 나타낸다.



① new.rlink ← pre.rlink; ~ ㉠
② new.rlink.llink ← new; ~ ㉡
③ pre.rlink ← new;
④ new.llink ← pre;

정답 : ④

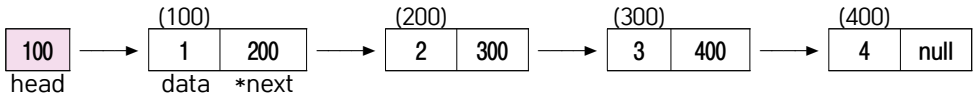
5. C 언어를 사용하여 연결리스트를 구현할 때, 관련이 없는 것은? [2010년 전산 7급]

① 비트 단위 논리곱
② 자기 참조 구조체
③ 동적 메모리 할당
④ 포인터

♤ C 언어를 사용한 연결리스트 – 자기 참조 구조체로 구현

// C에서 자기 참조 구조체는 다음처럼 선언해야 한다.

```
struct list
{
  int data;
  struct list *next;    //next는 반드시 포인터로 선언되어야 한다.
} node;
```



• 각 노드는 프로그램 실행 중 필요한 시점에 힙에 동적으로 메모리를 할당하게 된다.
• new = (struct node *)malloc(sizeof(struct node)); //힙에 동적 메모리 할당

정답 : ①