자료구조론	국가 전산 7급	2016년 8월 27일

♤ 합격선/최종합격인원(73.57점/27명) - 채용예정인원 26명 ♤

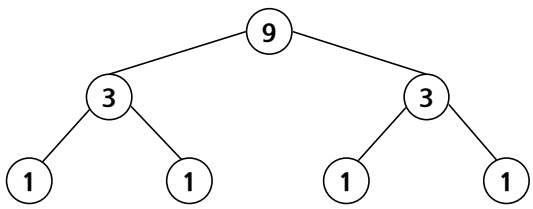
1. 다음 C 언어 함수를 이용하여 asterisk(9)를 수행할 때 출력되는 '∗'의 개수는?

```
-------------------------
#include <stdio.h>
void asterisk(int I)
{
  if(i>1){
    asterisk(i/3);
    asterisk(i/3);
  }
  printf("*");
}
-------------------------
```

① 6 ② 7
③ 8 ④ 9

♤ 재귀호출 - 재귀트리 이용

• 먼저, 주어진 재귀호출 문제를 재귀트리로 나타내면 다음과 같다.
• asterisk(9)를 수행



• 간단하게 답을 구하면, asterisk(9)가 7번 호출되므로 '∗'가 7개 출력된다.

• 주어진 문제처럼 같은 것을 단순하게 출력하는 것을 묻는 문제는 재귀트리를 그려서 푸는 것이 가장 빠르다.
• 전산 7급에서 재귀호출 문제는 매년 출제되고 있다.

정답 : ②

2. 가장 낮은 자릿수(LSD; Least Significant Digit) 우선 방식의 기수정렬(radix sort)을 이용하여, 다음과 같이 배열에 저장된 10진수 데이터를 오름차순으로 정렬하고자 한다. 오른쪽에서 두 번째 자릿수(digit) 기준의 정렬 단계 후에 위치 [3]에 있는 데이터로 옳은 것은? (단, 각 단계별 정렬의 기준이 되는 자릿수는 0에서 9까지의 값을 가진다)

위치	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
데이터	307	424	132	216	241	835	546	630

① 630 ② 546
③ 132 ④ 216

♤ 기수정렬

위치	원시리스트	Pass 1규	1차 재배열	Pass 2규	2차 재배열
[0]	307	0 : 630	630	0 : 307	307
[1]	424	1 : 241	241	1 : 216	216
[2]	132	2 : 132	132	2 : 424	424
[3]	216	3 :	424	3 : 630, 132, 835	**630**
[4]	241	4 : 424	835	4 : 241, 546	132
[5]	835	5 : 835	216	5 :	835
[6]	546	6 : 216, 546	546	6 :	241
[7]	630	7 : 307	307	7 :	546
		8 :		8 :	
		9 :		9 :	
		↓ 일의 자리값을 기준으로 분배		↓ 십의 자리값을 기준으로 분배	

◈ 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.

정답 : ①

3. 해시 테이블 HT의 크기는 11이고 0부터 10까지의 인덱스를 가진다. 해시함수 h(x)는 제산함수로서 h(x) =x mod 11로 표현된다. 충돌 해결책으로는 이차조사법(quadratic probing)을 사용한다. 해싱을 통해 해시 테이블에 저장할 레코드들의 키 값 순서는 다음과 같다. 마지막 레코드가 저장되는 해시 테이블 위치는? (단, 첫 번째 레코드가 저장되기 전에 해시 테이블은 비어 있고, 해시 테이블 버킷(bucket)당 슬롯(slot) 수는 1개이다)

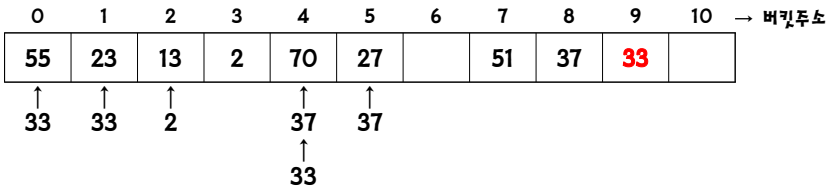
51, 27, 70, 55, 13, 2, 37, 23, 33

---

① HT[1] ② HT[4] ③ HT[6] ④ HT[9]

♤ 이차조사법

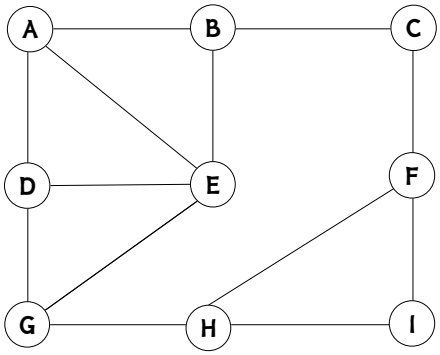
- 이차조사법은 충돌이 발생하면, 색인증가를 이차함수를 사용한다.
- 이차함수는 다양하지만, 주어진 문제에서 적용할 이차함수를 제시하지 않았다.
- 주어진 문제에서는 이차함수 x^2을 사용해야 답이 있다.(이의신청하면 될까?)
- 즉, 충돌이 발생하면 색인증가를 1, 4, 9, 16, … 순으로 적용해야 답이 있다.



h(51) = 51 mod 11 = 7	키 51은 버킷주소 7에 저장
h(27) = 27 mod 11 = 5	키 27은 버킷주소 5에 저장
h(70) = 70 mod 11 = 4	키 70은 버킷주소 4에 저장
h(55) = 55 mod 11 = 0	키 55는 버킷주소 0에 저장
h(13) = 13 mod 11 = 2	13은 버킷주소 2에 저장
h(2) = 2 mod 11 = 2	1차 충돌 발생, 색인증가를 1, 키 2는 버킷주소 3에 저장
h(37) = 37 mod 11 = 4	1차 충돌 발생, 색인증가를 1, 2차 충돌 발생, 색인증가를 4, 키 37은 버킷주소 8에 저장
h(23) = 23 mod 11 = 1	키 23은 버킷주소 1에 저장
h(33) = 33 mod 11 = 0	1차 충돌 발생, 색인증가를 1, 2차 충돌 발생, 색인증가를 4, 3차 충돌 발생, 색인증가를 9, 키 33은 버킷주소 9에 저장

정답 : ④

4. 시작 정점이 A일 때, 다음 그래프에 대한 깊이우선탐색(DFS: Depth First Search) 및 너비우선탐색(BFS: Breadth First Search)의 방문 순서로 옳은 것은? (단, 인접한 정점들은 알파벳 순서로 방문한다)



① DFS : A, B, C, F, H, G, D, E, I
BFS : A, B, C, D, E, G, F, H, I
② DFS : A, B, C, F, H, G, D, E, I
BFS : A, B, D, E, C, G, F, H, I
③ DFS : A, B, C, F, H, G, E, D, I
BFS : A, B, D, E, C, G, F, H, I
④ DFS : A, B, C, F, H, G, E, D, I
BFS : A, B, C, D, E, G, F, H, I

♤ **깊이우선탐색/너비우선탐색**

DFS	• 임의의 한 정점에서 방문되지 않은 **다른 정점**을 추적해 가는 방식이다. • 만약, 어떤 정점에서 모든 정점이 방문되었으면 되돌아가면서 방문되지 않은 정점을 찾는다.
BFS	• 임의의 한 정점에 방문되지 않은 **인접한 모든 정점**을 우선적으로 방문하는 방식이다. • 만약, 어떤 정점에서 모든 정점이 방문되었으면 가장 먼저 방문한 정점에서 똑 같이 인접한 정점 중에서 방문되지 않은 정점을 찾는다.

• DFS : A, B, C, F, H, G, D, E, I
• BFS : A, B, D, E, C, G, F, H, I

정답 : ②

5. 다음 C 프로그램 실행 결과로 첫 번째 줄에 출력되는 값과 두 번째 줄에 출력되는 값의 차이 값은? (단, int로 선언된 정수형 변수는 4바이트를 차지하고, char로 선언된 문자형 변수는 1바이트를 차지한다) [2016년 국가 7급]

```
---------------------------
#include <stdio.h>
void main() {
  struct node {
    int id;
    char name[8];
    int score;
  } A[10];
  printf("%d\n", &A[0]);
  printf("%d\n", &A[4]);
}
---------------------------
```

① 56　　② 64
③ 72　　④ 80

♤ **구조체 배열**

• 구조체 배열 원소 크기 = 4 + 8 + 4 = 16byte

A[0]	A[1]	A[2]	A[3]	A[4]	…	A[9]
100	116	132	148	164	…	← 주소

• 출력 값 차이 : &A[4]=164이고 &A[0]=100 이므로, 164 − 100 = **64**

정답 : ②