

2015년도 제52회 변리사 2차 국가자격시험 문제지

교 시	시험과목	시험시간	수험번호	성 명
2교시	데이터구조론	120분		

【 문제-1 】 (30점)

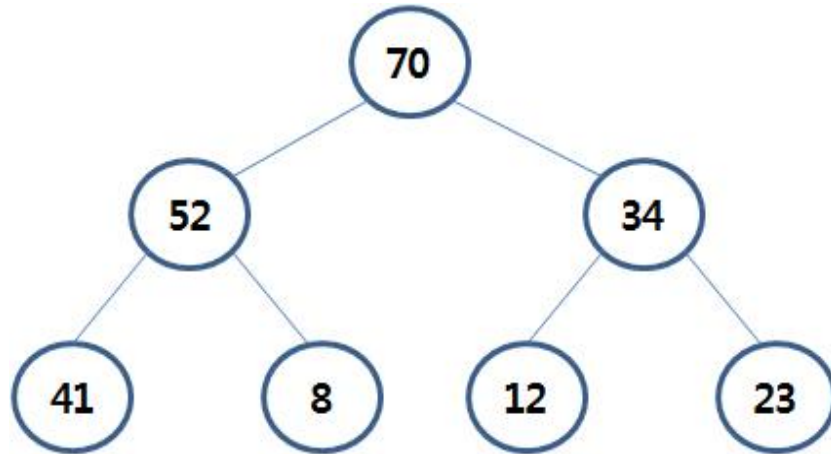
key 값이 각각 23, 42, 12, 56, 62, 79, 3인 입력 레코드가 순차적으로 삽입되고 해시함수가 $h(key) = key \bmod 11$ 로 정의된 상태에서 충돌(collision)이 발생하였을 경우 이를 해결하기 위한 충돌 해결법(collision resolution)에 관한 다음의 물음에 답하시오. (단, 아래와 같이 해시 테이블에 관한 배열의 크기는 11이고, 해시함수 $h(key)$ 에서 mod는 나머지 연산자이다.)

index	0	1	2	3	4	5	6	7	8	9	10
key											

- (1) 선형탐사(linear probing)를 수행한 결과를 위의 배열에 기록하고, 선형탐사의 문제점을 설명하시오. (6점)
- (2) 제곱탐사(quadratic probing)를 수행한 결과를 위의 배열에 기록하고, 제곱탐사의 문제점을 설명하시오. (9점)
- (3) 보조해시함수가 $h_2(key) = 7 - (key \bmod 7)$ 로 정의된 이중해싱(double hashing)을 수행한 결과를 위의 배열에 기록하고, 이중해싱의 문제점을 설명하시오. (9점)
- (4) 입력 레코드의 삽입이 계속되면 적재율(load factor)의 한계에 부딪혀 해시 테이블의 크기를 키워야 한다. 이 때 배치된 key 값을 재조정하는 작업을 수행하게 되는데, 재조정 작업의 수행시기를 두 가지만 열거하고 재조정하는 방법을 설명하시오. (6점)

【 문제-2 】 (20점)

우선순위 큐(priority queue)를 효율적으로 구현하기 위한 자료구조로서 힙(heap)을 이용할 수 있다. 다음 그림의 힙을 기준으로 물음에 답하시오. (단, 그림의 힙은 key 값이 큰 레코드가 우선순위가 높은 형태인 맥스 힙(max heap) 구조이고, 노드 안에 표시된 숫자는 레코드의 key 값을 의미한다.)



- (1) 위의 힙 구조를 다음의 배열로 표현할 때 빈칸에 적합한 숫자들을 채우시오. (4점)

index	0	1	2	3	4	5	6	7	8
key									

- (2) 위의 힙에 key 값이 60인 노드를 삽입한 후 힙의 모습을 복원하기 위한 업힙(up heap)이 수행된 결과를 (1)번에서 제시된 배열로 표현하시오. (7점)
- (3) (2)번 과정을 수행한 결과로 형성된 힙에서 노드를 삭제한 후 힙의 모습을 복원하기 위한 다운힙(down heap)이 수행된 결과를 (1)번에서 제시된 배열로 표현하고, 다운힙을 수행할 때 노드가 특정 위치에서 더 이상 아래 방향으로 다운되지 않고 멈추기 위해 만족되어야 하는 조건을 기술하시오. (9점)

【 문제-3 】 (30점)

알고리즘의 시간복잡도(time complexity)는 두 수를 비교하는 횟수로만 계산된다. 이 경우 하한값(lower bound)이란 주어진 문제를 푸는데 필요한 최소(smallest) 비교 횟수이다. 정렬이 되지 않은 서로 다른 n 개의 정수로 구성된 리스트에 관하여 다음의 물음에 답하시오.

- (1) 가장 큰 수를 구하는 문제에 관한 하한값이 $n-1$ 임을 설명하시오. (6점)
- (2) 가장 큰 수와 가장 작은 수를 같이 구하는데 필요한 비교 횟수를 함수 $f(n)$ 로 표현하고 그 이유를 설명하시오. (단, 이 문제에서 $n=2^i$ 라고 가정한다.) (12점)
- (3) 두 번째 큰 수를 구하는 문제에 관한 하한값을 함수 $f(n)$ 로, i 번째 큰 수를 구하는 문제에 관한 하한값을 함수 $g(n)$ 로 각각 표현하시오. (단, $3 \leq i \leq n$, i 는 정수) (6점)
- (4) (3)에서 구한 두 개의 하한값이 서로 같은지 혹은 다른지 여부를 판별하고, 그 이유를 설명하시오. (6점)

【 문제-4 】 (20점)

서로 다른 n 개의 정수로 구성된 리스트에서 어떤 정수 x 를 선형탐색(linear search)과 점프탐색(jump search)을 이용하여 탐색하려고 한다. 이때 점프탐색은 다음의 알고리즘에 기술된 절차로 탐색하는 방법이다. (단, 알고리즘의 시간복잡도(time complexity)는 두 수를 비교하는 횟수로만 계산된다.) 다음의 물음에 답하시오.

[알고리즘]

```
JUMP_SEARCH(list, lower, upper, jump, x) {  
  /* list는 오름차순으로 정렬되어 있음  
   x는 탐색하려는 정수, lower와 upper는 list의 탐색 범위를 지정하는 정수  
   jump는 list를 탐색할 때 건너뛰는 단위(jump size)  
   x의 위치가 발견되면 list 내의 위치를 return, 그렇지 않으면 0을 return  
   upper ≥ lower > 0  
   upper - lower + 1 ≥ jump ≥ 1 */  
  
  index = lower + jump - 1  
  while (upper > index) and (x > list[index])  
    index = index + jump  
  if upper > index  
    upper = index  
  LINEAR_SEARCH(list, index-jump+1, upper, x)  
  /* LINEAR_SEARCH는 선형탐색을 수행하는 함수  
   index-jump+1와 upper는 list의 탐색 범위를 지정하는 정수 */  
}
```

- (1) 리스트가 정렬되어 있지 않은 경우에 선형탐색을 수행하는 알고리즘에서 비교 횟수를 함수 $f(n)$ 로 표현하시오. (5점)
- (2) 리스트가 오름차순으로 정렬되어 있는 경우에 점프탐색을 수행하는 알고리즘에서 비교 횟수를 함수 $g(n, k)$ 로 표현하시오. (단, k 는 jump size) (7점)
- (3) 리스트가 정렬되어 있는 경우에 가장 효율적인 탐색 알고리즘을 제시하고 이 탐색 알고리즘의 비교 횟수를 함수 $h(n)$ 이라고 할 때, $h(n)$ 이 (2)번의 $g(n, k)$ 보다 탐색효율이 더 좋을 것을 시간복잡도를 이용하여 증명하시오. (8점)