

## <알고리즘 과제> 힙과 힙 정렬

문제: k-번째 작은 원소

- 1) 심층문제 6-6을 약간 확장한 과제이므로 원문제의 요구사항과 과제의 요구사항을 모두 충족해야 한다.
- 2) **중복이 있을 수 있는 n개의 정수들로 이루어진 무순리스트 L**을 구축하고 주어진 **k**에 대해 **k-번째 작은 원소를  $O(n + k \log n)$  시간에 찾아 인쇄하는 프로그램**을 작성하라.

프로그램 요구사항:

- 1) **findKthSmallest(L, n, k)**
  - 중복이 있을 수 있는 **n**개 정수 원소들로 이루어진 리스트 **L**의 원소 가운데 **k-번째로 작은 원소**를 찾아내 반환한다.
- 2) **buildList(n, min, max)**
  - 난수함수를 이용하여 **min ~ max** 사이의 중복이 있을 수 있는 정수 **n**개의 무작위, 무순리스트를 만들어 반환한다.
  - 리스트는 1D 배열 또는 단일연결리스트 가운데 하나를 자유롭게 선택하여 구현한다.
- 3) **main()**
  - **buildList(10, 1, 100)**을 호출하여 1과 100 사이의 정수 10개로 이루어진 리스트를 만들어 **L**에 저장한다.
  - 정수들 사이를 공백으로 구분하여, 리스트 **L**의 정수들을 한 라인에 출력한다.
  - **k = 1, 2, 3**을 차례로 인자로 사용하여 **findKthSmallest(L, 10, k)** 함수를 총 3회 호출한 결과를 모아 1개의 라인에 순서대로 출력한다.
  - **buildList(100000, 1, 1000000)**을 호출하여 1 ~ 100만 사이의 정수 10만개로 이루어진 리스트를 만들어 **L**에 저장한다.
  - 이번엔 생성된 리스트 **L**을 출력하지 않는다.
  - **k = 1, 100, 99900, 99999**를 차례로 인자로 사용하여 **findKthSmallest(L, 100000, k)** 함수를 총 4회 호출한 결과를 실행시간과 함께 4개의 라인으로 출력한다.
- 4) 위 함수들은 필히 작성해야 하며, 이외에도 **writeList** 등 필요하다고 판단되는 추가적인 함수들을 작성하라.

힌트:

- 1) 원문제의 실행시간 조건  $O(n + k \log n)$ 은 요구사항이기도 하지만 힌트도 된다는 점을 잘 생각해서 작성할 것.
- 2) Alg **main()**
  1. **L** ← **buildList(10, 1, 100)** {build a list of size 10}
  2. **writeList(L)** {write 10 elements in 1 line}
  3. for **k** ← 1 to 3 {mini test runs}  
**output[k] ← findKthSmallest(L, 10, k)**
  4. **write(output[1], output[2], output[3])** {write 3 elements in 1 line}
  5. **L** ← **buildList(100000, 1, 1000000)** {build a list of size 100,000}

```

6. karray ← {1, 100, 99900, 99999}    {an array of k values}
7. for k ← 0 to 3                        {main test runs}
    StartClock()
    e ← findKthSmallest(L, 100000, karray[k])
    StopClock()
    t ← CPUtime()
    write(e, t)                        {write e and t in 1 line}
8. return
    
```

**주의:**

- 1) 프로그램 전체를 통해 단 한 개의 **findKthSmallest** 함수만 작성하고 호출해야 함. 이를 어기면 요구 미충족으로 30% 감점.
- 2) **findKthSmallest**(**L**, **n**, **k**)의 인자 **L**은 항상 최근의 **buildList**에 의해 구축된 초기 리스트 **L**이어야 함. 이를 어기면 요구 미충족으로 30% 감점.
- 3) 위 명령문 3행 mini test runs의 결과값들이 부정확하면 부정확한 출력으로 30% 감점.
- 4) 요구한 실행시간  $O(n + k \log n)$ 에 미달하는 알고리즘은 요구 미충족으로 30% 감점. 이 실행시간은 **k**가 **n**에 근접한 경우를 제외하고는  $O(n \log n)$ 보다 일반적으로 빠르다. 좀더 쉽게 말해서, 작은 **k** 값에 대한 수행시간이 큰 **k** 값에 대한 수행시간보다 "상당히" 작지 않으면 요구 미충족 판정을 받게 됨.
- 5) **main**()을 실행하면 총 6개 라인을 인쇄해야 함. 이를 어기면 출력형식미비로 10% 감점.
- 6) 위 프로그램 요구사항에 제시된 함수명은 프로그램 내에서도 그대로 사용할 것. 이를 어기면 소스스타일 미비로 10% 감점.
- 7) 제 3자가 읽기에 쉽지 않다고 판단되는 코드 또는 코드 블록에는 주석을 붙일 것. 즉, 모든 함수와 프로그램의 데이터구조 및 알고리즘이 제 3자에 의해 충분히 이해될 수 있어야 함. 이를 어기면 소스스타일 미비로 10% 감점
- 8) 위 주의사항 및 감점사항은 반복 언급하지 않더라도 학기 중 모든 과제에 유사하게 적용될 것이며, 기타 감점사항은 강좌개요 ppt의 <과제채점기준>을 참고할 것.