



학교 공부/알고리즘

## [알고리즘](Python) 백준 2696번 중앙값 구하기 - 우선순위 큐

그리버 | 2023. 9. 19. 00:54 | 수정 | 삭제

유의사항	서유럽 4국 (영국/프랑스/스위스/이탈리아) 9~11일 #베스트셀러	[연합상품][지원특가]고성 당일 #해파랑길 #화진포 #아야진해변 #청간정
	4,280,800원~	32,900원~
	이탈리아 일주 8/9일 #가성비여행	장가계 4~6일 #100%출발확정
	1,699,000원~	419,000원~

## 문제

<https://www.acmicpc.net/problem/2696>

	2696번: 중앙값 구하기 www.acmicpc.net
--	-----------------------------------

주제	우선순위 큐
시간 / 메모리 제한	1 초 / 128 MB
정답 비율	49.190%

어떤 수열을 읽고,  
홀수 번째 수를 읽을 때 마다 지금까지 입력받은 값의 중앙값을 출력하는 프로그램을 작성하시오.

- 예를 들어 수열이 1, 5, 4, 3, 2
  - 홀수 번째 수는 1번째 수, 3번째 수, 5번째 수
  - 1번째 수를 읽었을 때 중앙값은 1, 3번째 수를 읽었을 때 4, 5번째 수를 읽었을 때 3

## 입력

- 첫째 줄에 테스트 케이스의 개수( $1 \leq T \leq 1,000$ )가 주어진다.

- 각 테스트 케이스의 첫째 줄에는 수열의 크기  $M$  ( $1 \leq M \leq 9999$ ,  $M$ 은 홀수)이 주어지고
- 그 다음 줄부터 이 수열의 원소가 차례대로 주어진다.
- 원소는 한 줄에 10개씩 나누어져있고, 32비트 부호있는 정수

## 출력

- 각 테스트 케이스에 대해
  - 첫째 줄에 출력하는 중앙값의 개수를 출력
  - 둘째 줄에 홀수 번째 수를 읽을 때마다 구한 중앙값을 차례대로 공백으로 구분해 출력
  - 이때, 한 줄에 10개씩 출력해야 함

- 예제 입력 1

```
3
9
1 2 3 4 5 6 7 8 9
9
9 8 7 6 5 4 3 2 1
23
23 41 13 22 -3 24 -31 -11 -8 -7
3 5 103 211 -311 -45 -67 -73 -81 -99
-33 24 56
```

- 예제 출력 1

```
5
1 2 3 4 5
5
9 8 7 6 5
12
23 23 22 22 13 3 5 5 3 -3
-7 -3
```

## 우선순위 큐 구현

우선순위 큐는 키를 가진 원소 집합을 위한 자료 구조로 최대 우선순위 큐와 최소 우선순위 큐가 있다.

우선순위 큐는 이진 탐색 트리 또는 힙으로 구현할 수 있지만 힙으로 구현하는 경우가 많다.

파이썬에서는 최소 힙을 구현할 수 있는 `heapq` 모듈을 제공하므로 이를 이용해 우선순위 큐를 구현할 수 있다.

최소 우선순위 큐를 구현해보자.

## 우선순위 큐의 ADT

```
### 우선순위 큐의 ADT ###
# Object
: 키를 가진 원소 집합
# Operation
1. is_empty() -> Boolean
   : 큐가 비어있으면 True, 아니면 False 반환
2. push(item)
   : 큐에 원소 item 삽입
3. pop() -> element
   : 큐에서 키 값이 가장 작은 요소를 삭제하며 반환
4. min() -> element
   : 큐에서 키 값이 가장 작은 요소를 반환
```

## 우선순위 큐 구현

```
from heapq import heappush, heappop
```

```

class Element:

    def __init__(self, key, string):
        self.key = key
        self.data = string

class MinPriorityQueue:

    def __init__(self):
        self.heap = []

    def is_empty(self):
        if not self.heap:
            return True
        return False

    def push(self, item):
        heappush(self.heap, (item.key, item.data))

    def pop(self):
        return heappop(self.heap)

    def min(self):
        return self.heap[0]

```

## 문제 풀이

### 시간 초과된 풀이

```

import sys
import heapq
import copy

T = int(input())

for _ in range(T):
    heap = []

    M = int(input())
    nums = []
    for _ in range(M // 10 + 1):
        nums += list(map(int, input().split()))

    print((M + 1) // 2)
    for m in range(M): # m = 0~8

        heapq.heappush(heap, nums[m])
        if (m + 1) % 2 == 1: # m+1 = 1, 3, 5
            temp = copy.deepcopy(heap)

            for _ in range((m + 2) // 2): # m+2 = 2, 4, 6
                mid = heapq.heappop(temp)

            print(mid, end=' ')
            if (m != 0 and m % 9 == 0) or m == M-1:
                print()

```

공감

'학교 공부 > 알고리즘' 카테고리의 다른 글

[알고리즘](Python) 트리 (1)	2023.09.12
[알고리즘](Python) 백준 10828번 - 스택 (0)	2023.09.12
[알고리즘](Python) 백준 10845번 - 큐 (0)	2023.09.12
[알고리즘](Python) 리스트 개념과 그림으로 보는 파이썬 코드 구현 (0)	2023.09.11

'학교 공부/알고리즘' Related Articles



[알고리즘](Python) 트리



[알고리즘](Python) 백준 10828번 - 스택



[알고리즘](Python) 백준 10845번 - 큐



[알고리즘](Python) 리스트 개념과 그림으로 보는 파이썬...

☐ Secret

안녕하세요! 어떤 댓글이든 환영합니다! 🍷

댓글달기