

# Week 9

- 주의 사항: 부정행위 금지, STL 사용 금지(특히, algorithm 사용금지), 인터넷 금지, STL 벡터를 이용하여 구현할 것.
- 표준 입출력 사용을 권장 (C는 scanf / printf, C++은 cin / cout)

## 문제 1

정렬된 벡터로 구현된 우선순위 큐  $P$ 를 이용하여, 자연수  $n$  ( $n$ 은  $1 \leq n \leq 10,000$ )으로 구성된 수열  $S$ 를 입력 받으면, 삽입 정렬을 사용하여 오름차순으로 정렬된 수열  $S$ 를 출력하고 각 정렬 단계별 처음과 마지막 원소의 합 (아래 표 내 수열  $S$ 의 단계2의 빨간색 글씨 참조)을 구하는 프로그램을 구현한다. (여기서 정렬 단계란 우선순위 큐  $P$ 에 원소를 다 입력한 후 수열  $S$ 에서 정렬을 수행하는 것을 의미한다.) 단, 이때 수열  $S$ 의 단계2 내 첫 번째 단계는 값이 하나 밖에 없으므로 연산 대상에서 제외한다.

예시로 수열  $S=(7, 4, 8, 2, 5, 3, 9)$ 를 입력 받았다고 가정하자.

	수열 $S$	우선순위 큐 $P$
입력	(7, 4, 8, 2, 5, 3, 9)	( )
단계1 (a)	(4, 8, 2, 5, 3, 9)	(7)
(b)	(8, 2, 5, 3, 9)	(4, 7)
(c)	(2, 5, 3, 9)	(4, 7, 8)
(d)	(5, 3, 9)	(2, 4, 7, 8)
(e)	(3, 9)	(2, 4, 5, 7, 8)
(f)	(9)	(2, 3, 4, 5, 7, 8)
(g)	( )	(2, 3, 4, 5, 7, 8, 9)
단계2 (a)	(2)	(3, 4, 5, 7, 8, 9)
(b)	(2, 3)	(4, 5, 7, 8, 9)
(c)	(2, 3, 4)	(5, 7, 8, 9)
(d)	(2, 3, 4, 5)	(7, 8, 9)
(e)	(2, 3, 4, 5, 7)	(8, 9)
(f)	(2, 3, 4, 5, 7, 8)	(9)
(g)	(2, 3, 4, 5, 7, 8, 9)	( )

단계1에서 수열  $S$ 의 첫번째 원소를 반복적으로 삭제하여 우선순위 큐  $P$ 에 삽입한다.

단계2에서는 우선순위 큐  $P$ 의 최솟값을 반복적으로 삭제하여 시퀀스  $S$ 의 끝에 삽입한다.

최종적으로 정렬된 수열 2 3 4 5 7 8 9과 정렬 단계에서의 처음 및 마지막 원소의 합인 5 6 7 9 10 11을 출력한다.

## 입력

1. 첫 번째 줄에 테스트 케이스의 개수  $T$  ( $1 \leq T \leq 1,000$ )을 입력한다.
  - 1) 각 케이스마다 다음이 반복되어 나타난다.
  - 2) 각 케이스의 첫 번째 줄에 입력 받을 자연수의 개수  $N$  ( $5 \leq N \leq 20$ )을 입력한다.

- 3) 각 케이스의 두 번째 줄에  $N$ 개의 자연수  $n$  ( $1 \leq n \leq 10,000$ )를 띄어쓰기로 구분하여 입력한다.  
(단, 각 케이스마다 입력되는 자연수  $n$ 은 중복되지 않는다.)

## 출력

각 케이스마다 입력 받은 수열을 정렬한 수열과 첫 번째 단계를 제외한 각 정렬 단계의 처음 및 마지막 원소의 합의 수열을 각각 개행하여 출력한다.

## 예제 입출력

예제 입력	예제 출력
8	9 23 30 75 84 94 100
7	32 39 84 93 103 109
9 94 75 30 23 100 84	3 30 34 42 47 55 58 61 72 90 96
11	33 37 45 50 58 61 64 75 93 99
72 55 58 61 96 34 47 30 3 90 42	9 12 17 24 27 53 63 77 81
9	21 26 33 36 62 72 86 90
12 81 63 53 9 77 24 27 17	3 41 59 65 81
5	44 62 68 84
65 41 3 81 59	5 21 40 41 42 47 48 63 65 72 78 85 92 96
14	26 45 46 47 52 53 68 70 77 83 90 97 101
92 5 85 48 41 21 72 47 40 65 96 42 63 78	15 32 46 47 51 54 59 69 86 94
10	47 61 62 66 69 74 84 101 109
54 46 47 69 15 94 86 32 51 59	6 10 32 59 79 91 92 97
8	16 38 65 85 97 98 103
91 32 59 10 6 97 92 79	3 14 27 33 38 49 61 62 63 65 68 77 93
13	17 30 36 41 52 64 65 66 68 71 80 96
33 68 62 3 14 61 27 38 63 49 93 77 65	