

## 4주차 주간학습보고서(4/12~4/18)

분류 1	전공		
팀 명	정웅팡팡	제출자	김도연
학습 일시	2021.04.12 17:00 ~ 2021.04.12 19:00	마감 일시	2021.04.12 19:00
제출 일시	2021.04.13 11:20	수정 일시	2021.04.13 11:20

[2021-1 광운스터디그룹]

# 주간학습보고서

학습주제	정렬
학습장소	기념관 로비
학습방법 및 학습개요	<p>정렬 알고리즘</p> <p>1. 정렬의 종류</p> <ul style="list-style-type: none"> <li>- bubble sort</li> <li>- merge sort</li> </ul> <p>2. 정렬 문제 (백준 알고리즘: <a href="https://www.acmicpc.net/">https://www.acmicpc.net/</a>)</p> <ul style="list-style-type: none"> <li>- 10989(수 정렬3, 실버 5)</li> <li>- 1181(단어 정렬, 실버5)</li> <li>- 2108(통계학, 실버4)</li> <li>- 11497(통나무 건너뛰기, 실버1)</li> <li>- 5052(전화번호 목록, 골드4)</li> <li>- 1377(버블소트, 골드1)</li> </ul> <p>3. 정리</p> <p>깃허브 인증</p>
다음 주 계획	<p>주제: 완전 탐색</p> <p>일시: 2021.04.19.</p> <p>장소: 기념관 2층</p> <p>보고서 제출자: 이윤희</p> <p>과제:</p> <ul style="list-style-type: none"> <li>- 1018번: 체스판 다시 칠하기 (실버 5)</li> <li>- 9663번: <a href="#">N-Queen</a> (골드 5)</li> <li>- 12100번: 2048 (골드 2)</li> </ul>

\* 학습개요의 내용을 구체적으로 작성해 주세요.

<학습 내용>

1. 정렬의 종류

- **bubble sort**: 서로 이웃한 데이터들을 비교하며 가장 큰 데이터를 가장 큰 데이터의 가장 뒤로 보내며 정렬하는 방식이다.

성능:  $O(n^2)$

- **merge sort**: 하나의 리스트를 정렬하기 위해서 해당 리스트를 n개의 서브리스트로 분할하여 각각을 정렬한 후, 정렬된 n개의 서브리스트로 합병시켜서 정렬시키는 방법

성능:  $O(n \log n)$

2. 정렬 문제

1) 10989(수 정렬3, 실버 5)

## 문제

N개의 수가 주어졌을 때, 이를 오름차순으로 정렬하는 프로그램을 작성하시오.

- 풀이(이윤희)

```
N = int(input())
c=[0] * 10000

for _ in range(N):
    c[int(input())-1] += 1

for i in range(len(c)):
    if c[i] > 0:
        for _ in range(c[i]):
            print(i+1)
```

2) 1181(단어 정렬, 실버 5)

## 문제

알파벳 소문자로 이루어진 N개의 단어가 들어오면 아래와 같은 조건에 따라 정렬하는 프로그램을 작성하시오.

1. 길이가 짧은 것부터
2. 길이가 같으면 사전 순으로

- 풀이 (안희승)

```
1  import sys
2  n = int(sys.stdin.readline())
3  word = set([])
4
5  for i in range(n):
6      word.add(sys.stdin.readline().strip())
7
8  word = list(word)
9  word.sort()
10 word.sort(key = lambda x:len(x))
11
12 for i in word:
13     print(i)
```

3) 2108 (통계학, 실버 4)

## 문제

알파벳 소문자로 이루어진 N개의 단어가 들어오면 아래와 같은 조건에 따라 정렬하는 프로그램을 작성하시오.

1. 길이가 짧은 것부터
2. 길이가 같으면 사전 순으로

```

1  import sys
2  from collections import Counter
3
4  input = sys.stdin.readline
5  n = int(input())
6  li = []
7  for _ in range(n):
8      li.append(int(input()))
9  li.sort()
10
11  def mode(x):
12      mode_dict = Counter(x)
13      modes = mode_dict.most_common()
14      if len(x) > 1:
15          if modes[0][1] == modes[1][1]:
16              mod = modes[1][0]
17          else:
18              mod = modes[0][0]
19      else:
20          mod = modes[0][0]
21      return mod
22
23  # 산술평균
24  print(round(sum(li)/n))
25  # 중앙값
26  print(li[n//2])
27  # 최빈값
28  print(mode(li))
29  # 범위
30  print(li[-1] - li[0])

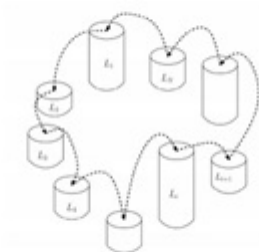
```

- 풀이(김도연)

#### 4) 11497(통나무 건너뛰기, 실버1)

##### 문제

남규는 통나무를 세워 놓고 건너뛰기를 좋아한다. 그래서 N개의 통나무를 원형으로 세워 놓고 뛰어놀려고 한다. 남규는 원형으로 인접한 열 통나무로 건너뛰는데, 이때 각 인접한 통나무의 높이 차이가 최소가 되게 하려 한다.



통나무 건너뛰기의 난이도는 인접한 두 통나무 간의 높이의 차의 최댓값으로 결정된다. 높이가 {2, 4, 5, 7, 9}인 통나무들을 세우려 한다고 가정하자. 이를 {2, 9, 7, 4, 5}의 순서로 세웠다면, 가장 첫 통나무와 가장 마지막 통나무 역시 인접해 있다. 즉, 높이가 2인 것과 높이가 5인 것도 서로 인접해 있다. 배열 {2, 9, 7, 4, 5}의 난이도는  $|2-9| = 7$ 이다. 우리는 더 나은 배열 {2, 5, 9, 7, 4}를 만들 수 있으며 이 배열의 난이도는  $|5-9| = 4$ 이다. 이 배열보다 난이도가 낮은 배열은 만들 수 없으므로 이 배열이 남규가 찾는 답이 된다.

- 풀이 (김도연)

```
1 # https://velog.io/@ready2start/Python-%E8%9B%B3%E6%A0%B7%E6%9C%9F-11497-%ED%9A%B3%E6%9D%B2%E6%AC%B4-%EA%B1%B4%E6%9B%B2%E6%9A%B0
2 from sys import stdin
3
4 t = int(stdin.readline())
5
6 for _ in range(t):
7     n = int(stdin.readline())
8     heights = [int(x) for x in stdin.readline().split()]
9     heights.sort()
10
11     max_level = 0
12     for i in range(2, n):
13         max_level = max(max_level, abs(heights[i] - heights[i - 2]))
14
15 print(max_level)
```

merge sort의 정의: 문제를 작은 2개의 문제로 분리하고 각각을 해결한 다음, 결과를 모아서 원래의 문제를 해결하는 전략  
=> swap(바꾸는 횟수)은 동일하지만 시간이 더 적게 걸린다.

```

1  import sys
2  input=sys.stdin.readline
3
4  def merge(a,b):
5      global cnt
6      la,lb=len(a),len(b)
7      i,j=0,0
8      temp=[]
9      while i<la and j<lb:
10         if a[i]>b[j]:
11             temp.append(b[j])
12             j+=1
13             cnt+=la-i
14         else:
15             temp.append(a[i])
16             i+=1
17     if i==la:
18         temp.extend(b[j:])
19     else:
20         temp.extend(a[i:])
21     return temp
22
23 def merge_sort(arr):
24     if len(arr)<=1:
25         return arr
26     left=0
27     right=len(arr)-1
28     mid=(left+right)//2
29     return merge(merge_sort(arr[left:mid+1]),merge_sort(arr[mid+1:]))
30
31 n=int(input())
32 cnt=0
33 arr=list(map(int,input().split()))
34 merge_sort(arr)
35 print(cnt)

```

<https://cantcoding.tistory.com/33> 참고

\* 학습활동 사진(필수)

온라인 활동시 : 시작 화면, 종료 화면(구성원 모두 나온 화면캡처)

오프라인 활동시 : 현장 활동 사진(구성원 및 활동 시간이 나오도록 촬영)

