

스터디 주간 활동 보고서

팀명	별꿀오소리	제출자 성명	서준원
참여 명단	서준원, 정찬원, 임소정, 강인우, 정민섭, 송주훈		
모임 일시	2025 년 02 월 06 일 21 시 ~ 22 시 10 분(1 시간 10 분)		
장소	온라인(구글미트)	출석 인원	5 / 6 송주훈님 개인 사정으로 화상미팅 미참여 (스터디 활동에는 참여)
학습목표	<p>백준 알고리즘 문제를 파이썬을 사용하여 풀어보며 문제 해결 능력 향상</p> <p>선정 문제 :</p> <ol style="list-style-type: none">1. 백준 1463 번 1 로 만들기 s32. 백준 1051 번 숫자 정사각형 s33. 백준 1967 번 트리의 지름 g44. 백준 32982 약물복용 s2		

	<p>5. 백준 1541 잃어버린 괄호 s2</p> <p>6. 백준 2869 번 달팽이는 올라가고 싶다 b1</p> <p>7. 백준 10814 나이순 정렬 s5</p>
<p>학습내용</p>	<p>1. 백준 1463 번 1로 만들기 s3(송주훈 선정 문제)</p> <p>https://www.acmicpc.net/problem/1463</p> <p>송주훈 풀이</p> <p>다이나믹 프로그래밍을 통해 문제를 해결합니다.</p> <p>DP 다이나믹 프로그래밍 : 복잡한 문제를 작은 문제로 나누어 해결하는 알고리즘 기법</p> <p>재귀는 top->down 방식으로 접근하나 다이나믹 프로그래밍은 bottom->up 의 방식으로 문제를 해결합니다.</p> <p>위의 문제를 해결하는 경우는 정수 1 부터 N 까지 리스트를 만들고 그 값들을 이용해 올라가는 bottom up 방식으로 해결합니다.</p> <ol style="list-style-type: none"> 1. 먼저 0 으로 초기화되고 길이가 최대 인풋값인 10 의 6 승으로된 리스트 d 를 생성합니다 2. d[0] d[1]의 값은 0 으로 설정합니다. 0 은 input 으로 사용할 수 없고, 1 은 1 로 만드는데에 연산이 필요없으니 0 으로 합니다. 3. d[2]부터는 다음과 같은 연산을 비교합니다 <ul style="list-style-type: none"> - d[i]값을 구하기 위해 d[i-1]값에 1 더하기 - d[i]가 2 나 3 으로 나뉘 떨어질 경우 $d[i/2] + 1$ 이나 $d[i/3] + 1$ 값을 위에서 구한 $d[i-1] + 1$ 의

값과 비교하기

- 비교한 값 중 최솟값이 d[i]의 값이 됨

4. 이러한 연산 과정을 반복하여 d[n]값을 구하면 정답이 나옵니다.

```
1  n =int(input())
2
3  d =[0]*1000001
4
5  for i in range(2, n+1):
6      d[i] =d[i-1] + 1
7      if i%2 == 0:
8          d[i] =min(d[i], d[i//2] + 1)
9      if i%3 == 0:
10         d[i] =min(d[i], d[i//3] + 1)
11
12 print(d[n])
```

후기) 다이나믹 프로그래밍의 개념을 파악하고 연습문제를 풀며 개념을 익힐 수 있었습니다. 다른 다이나믹 프로그래밍 문제를 풀어보며 알고리즘 기법에 더 익숙해져야겠다고 생각했습니다!

추가로 저번 주 문제인 2839 번 설탕 배달 문제를 통해 복습해볼 수 있었습니다.

2. 백준 1051 번 숫자 정사각형 s3(정찬원 선정 문제)

<https://www.acmicpc.net/problem/1051>

정찬원 풀이

```

1  # 문제
2  # NxM크기의 직사각형이 있다. 각 칸에는 한 자리 숫자가 적혀 있다.
3  # 이 직사각형에서 꼭짓점에 쓰여 있는 수가 모두 같은 가장 큰 정사각형을 찾는 프로그램을 작성하시오.
4  # 이때, 정사각형은 행 또는 열에 평행해야 한다.
5
6  # 입력
7  #첫째 줄에 N과 M이 주어진다. N과 M은 50보다 작거나 같은 자연수이다.
8  #둘째 줄부터 N개의 줄에 수가 주어진다.
9
10 #출력
11 #첫째 줄에 정답 정사각형의 크기를 출력한다.
12
13 #입력
14 n, m = map(int, input().split()) #직사각형의 행(n)과 열(m) 입력받기
15 square = [input().strip() for _ in range(n)] #둘째 줄부터 n개의 줄에 수를 리스트로 입력받기
16
17 min_square_size = 1 # 가장 작은 정사각형의 크기 1로 초기화
18
19 for i in range(n): #i(행), j(열)는 정사각형의 좌측 상단 꼭짓점 좌표
20     for j in range(m):
21         for size in range(1, min(n-i, m-j)): #size++ ->정사각형 한변의 길이를 늘려가면서 가능한 정사각형 탐색
22             #min 함수를 통해 현재 위치에서 만들 수 있는 최대 정사각형 크기 제한
23             #정사각형 4개의 꼭짓점 값이 같은지 확인하기
24             if (square[i][j] == square[i][j+size]
25                 == square[i+size][j] == square[i+size][j+size]):
26                 #꼭짓점 네 개 값 같으면 정사각형 생성
27                 min_square_size = max(min_square_size, (size+1)**2)
28
29 print(min_square_size)
30

```

3. 백준 1967 번 트리의 지름 g4(임소정 선정 문제)

<https://www.acmicpc.net/problem/1967>

임소정 풀이

```

Code  Blame  88 lines (53 loc) · 3.88 KB
1  # 문제
2  # 트리(tree)는 사이클이 없는 무방향 그래프이다. 트리에서는 어떤 두 노드를 선택해도 둘 사이에 경로가 항상 하나만 존재하게 된다. 트리에서 어떤 두 노드를 선택해서 양쪽으로 꼭 닿길 때, 가장 깊게 들어나는 경우가 있을 것이다. 이를 때 트리의 모든 노
3
4
5
6  # 어떤 두 노드 사이의 경로의 길이를 트리의 지름이라고 한다. 정확히 정의하자면 트리에 존재하는 모든 경로를 중에서 가장 긴 것의 길이를 말한다.
7
8  # 입력으로 후트가 있는 트리를 가중치가 있는 간선들로 줄 때, 트리의 지름을 구해서 출력하는 프로그램을 작성하시오. 입력과 같은 트리가 주어진다면 트리의 지름은 45가 된다.
9
10
11
12  # 트리의 노드는 1부터 n까지 번호가 매겨져 있다.
13
14  # 입력
15  # 파일의 첫 번째 줄은 노드의 개수 n(1 ≤ n ≤ 10,000)이다. 둘째 줄부터 n-1개의 줄에 각 간선에 대한 정보가 들어온다. 간선에 대한 정보는 세 개의 정수로 이루어져 있다. 첫 번째 정수는 간선의 연결하는 두 노드 중 부모 노드의 번호를 나타내고, 두 번째
16
17  # 출력
18  # 첫째 줄에 트리의 지름을 출력한다.
19
20  #####get 도움..
21
22
23  #####입시문제 제공 코드
24
25  '''
26  def my_dfs(graph, start_node):
27      stack = [] # DFS는 스택을 사용
28      visited = [] # 방문한 노드를 저장
29
30      stack.append(start_node) # 시작 노드를 스택에 추가
31
32      while stack:
33          node = stack.pop() # 스택에서 pop (후입전출 LIFO)
34
35          if node not in visited:
36              visited.append(node) # 방문 처리
37              stack.extend(graph[node]) # 현재 노드의 연결 노드들을 추가
38
39      print("dfs - ", visited)
40      return visited
41
42  '''
43
44
45  '''
46  def my_bfs(graph, start_node):
47      stack = []
48      visited = [node: 1 for node in graph] # 방문 여부 및 거리 저장
49      stack.append((start_node, 0)) # 스택에 (노드, 누적거리) 저장
50      visited[start_node] = 0 # 시작 노드는 거리 0
51

```

```

50     farthest_node = start_node
51     max_distance = 0
52
53     while stack:
54         node, dist = stack.pop()
55         #스택 내의 마지막 (노드, 거리)
56
57         if dist > max_distance:
58             max_distance = dist
59             farthest_node = node
60
61         for neighbor, weight in graph[node]:
62             # graph邻接表를 역순대로 방문
63             if visited[neighbor] == -1:
64                 # 방문하지 않았으면 아직 거리가 -1
65                 visited[neighbor] = dist + weight
66                 stack.append((neighbor, dist + weight))
67
68     return farthest_node, max_distance
69
70 # 입력
71 n = int(input().strip())
72 graph = [[] for i in range(1, n + 1)]
73
74 for _ in range(n - 1):
75     parent, child, weight = map(int, input().split())
76     graph[parent].append((child, weight))
77     graph[child].append((parent, weight))
78     # 무방향 그래프(==양방향)
79
80 # 1. 루트(1번 노드)에서 가장 먼 노드 찾기
81 farthest_node, _ = my_dfs(graph, 1)
82
83 # 2. 가장 먼 노드에서 다시 dfs 수행하여 최대의 치를 찾기 <-----이게 진짜!
84 _, tree_diameter = my_dfs(graph, farthest_node)
85
86 print(tree_diameter)
87
88

```

4. 백준 32982 약물복용 s2(강인우 선정 문제)

<https://www.acmicpc.net/problem/32982>

강인우 풀이

[아이디어]

식사 시간을 '시작 시간', '끝 시간'으로 구분하고 약효 지속 시간을 담은 변수 per를 선언

1. per보다 다음 식사 시작 시간이 더 멀리 있으면 약물 복용 불가
2. 약물이 복용 가능하다면 때는 최대한 늦게 먹어야됨 → 지속시간과 식사 끝 시간 중 작은값을 선택
다음 지속 시간은 $k + \min(\text{per}, \text{끝 시간})$
3. 저녁 시간 이후 per를 구할 때는 1440분을 빼서 다음 날에 대한 지속 시간을 구해야 함.

```

n, k = map(int, input().split())
lst = tuple(map(int, input().split()))

per = lst[1] + k

flag = True
for i in range(n):
    for j in range(3): #아침 점심 저녁: 01 / 23 / 45
        if per < lst[2*j]:
            flag = False
            break

        per = k + min(per, lst[2*j+1])

    if j==2: #저녁->아침 넘어갈때
        per -= 1440

    if not flag:
        break

print('YES') if flag else print("NO")

```

5. 백준 1541 잃어버린 괄호 s2(정민섭 선정 문제)

<https://www.acmicpc.net/problem/1541>

정민섭 풀이

```
1  arr = input().split('-')
2  m = 0
3
4  for i in arr[0].split('+'):
5      m += int(i)
6
7  for i in arr[1:]:
8      for j in i.split('+'):
9          m -= int(j)
10
11 print(m)
```

6. 백준 2869 번 달팽이는 올라가고 싶다 b1(서준원 선정 문제)

<https://www.acmicpc.net/problem/2869>

서준원 풀이

	A	B	V
예시)	5	1	15

	첫 걸음	남는 걸음
1일차	3m	4m
2일차	9m	8m
3일차	13m	12m
4일차	15m	

→ 4일 동안에 정상 도착 = 4일차!

$$\frac{V-B}{A-B} = \frac{15-1}{5-1} = \frac{14}{4} = 3.5 \xrightarrow{\text{소수점 올림}} 4\text{일}$$

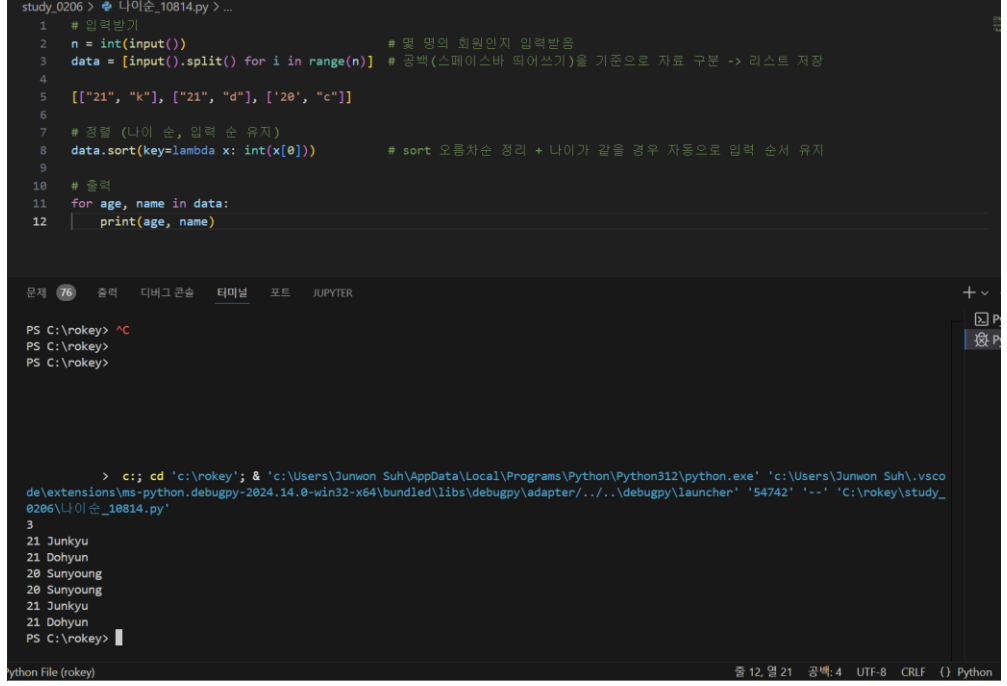
```
import math
```

```
A, B, V = map(int, input().split())
days = math.ceil((V - B) / (A - B))
print(days)
```

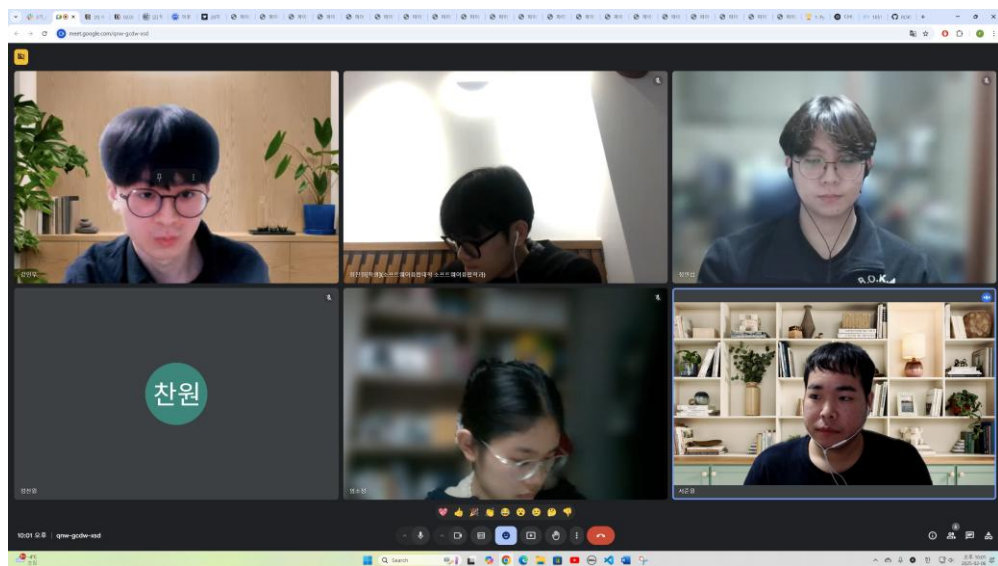
7. 백준 10814 나이순 정렬 s5(서준원 선정 문제)

<https://www.acmicpc.net/problem/10814>

서준원 풀이

	 <pre> study_0206 > 나이순_10814.py > ... 1 # 입력받기 2 n = int(input()) # 몇 명의 회원인지 입력받음 3 data = [input().split() for i in range(n)] # 공백(스페이스바 찍어쓰기)을 기준으로 자료 구분 -> 리스트 저장 4 5 [["21", "k"], ["21", "d"], ["20", "c"]] 6 7 # 정렬 (나이 순, 입력 순 유지) 8 data.sort(key=lambda x: int(x[0])) # sort 오름차순 정렬 + 나이가 같을 경우 자동으로 입력 순서 유지 9 10 # 출력 11 for age, name in data: 12 print(age, name) </pre> <pre> PS C:\rokey> ^C PS C:\rokey> PS C:\rokey> > c:; cd 'c:\rokey'; & 'c:\Users\Junwon Suh\AppData\Local\Programs\Python\Python312\python.exe' 'c:\Users\Junwon Suh\.vsco de\extensions\ms-python-debugpy-2024.14.0-win32-x64\bundled\libs\debugpy\adapter\..\..\debugpy\launcher' '54742' '--' 'C:\rokey\study_ 0206\나이순_10814.py' 3 21 Junkyu 21 Dohyun 20 Sunyoung 20 Sunyoung 21 Junkyu 21 Dohyun PS C:\rokey> </pre> <p>Python File (rokey) 줄 12, 열 21 공백: 4 UTF-8 CRLF () Python</p>
<p>활동평가</p>	<p>지난번 문제 난이도 조정에 어려움을 겪었던 것을 반영하여 난이도 재조정 완료. 다가올 시험의 난이도에 대해서 서로 의견 나눔. 이번주도 성공적인 스터디 완료.</p>
<p>과제</p>	<p>~2/11 (화) : 시험 대비(우리 모두 파이팅!)</p> <p>~2/13 (목) : 파이토치에 대해서 알아오기</p>
<p>향후 계획</p>	<p>2.13 스터디 주제: PyTorch 사용법</p>

첨부 자료



https://github.com/ChanwonJung/ROKEY_team6_study/tree/main