

두산 Rokey Boot Camp

스터디 주간 활동 보고서

팀명	Robo:Loop	제출자 성명	홍송은
참여 명단	전효재, 장수근, 홍송은, 김사웅		
모임 일시	2025년 04월 01일 17시 00분 ~ 18시 00분(총 1시간)		
장소	온라인 구글 미팅	출석 인원	4
학습목표	<ul style="list-style-type: none">DFS의 개념과 재귀 및 스택을 활용한 구현 방식을 이해한다.문제 상황에서 DFS를 적용해 경우의 수를 탐색하는 능력을 기른다.실제 문제 풀이를 통해 DFS 응용력과 문제 해결 능력을 향상시킨다.		
학습내용	<p>1. DFS(깊이 우선 탐색) 알고리즘 학습</p> <ul style="list-style-type: none">그래프나 트리에서 한 방향으로 끝까지 탐색한 후, 다시 돌아와 다른 경로 탐색재귀 or 스택 사용해서 구현모든 조합이나 경우의 수를 탐색해야 하는 문제에 적합 <p>2. DFS 알고리즘 문제 풀이 - 프로그래머스</p> <ul style="list-style-type: none">타켓 넘버 문제(홍송은) <p>: 정수 배열 numbers가 주어졌을 때, 각 숫자 앞에 + 또는 -를 붙여 연산 결과가 target이 되는 경우의 수를 구하는 문제</p>		

- 순서를 바꾸지 않고 계산해야 하므로 DFS가 적합한 방식임을 판단. 만약 순서를 바꿔도 된다면 순열도 함께 고려해야 함을 이해함.
- 각 숫자에 + 또는 -를 붙이는 모든 경우를 트리 구조로 DFS를 활용하여 탐색. 직접 트리를 그리며 이해를 돕기 위한 자세한 주석을 추가.
- 재귀 호출로 누적합을 계산하며, leaf node에서 target과 일치하면 count 증가.

- 아이템 줍기 문제(전효재)

: 여러 개의 이어진 사각형이 주어졌을때 가장 밖 테두리의 두 점 최단거리

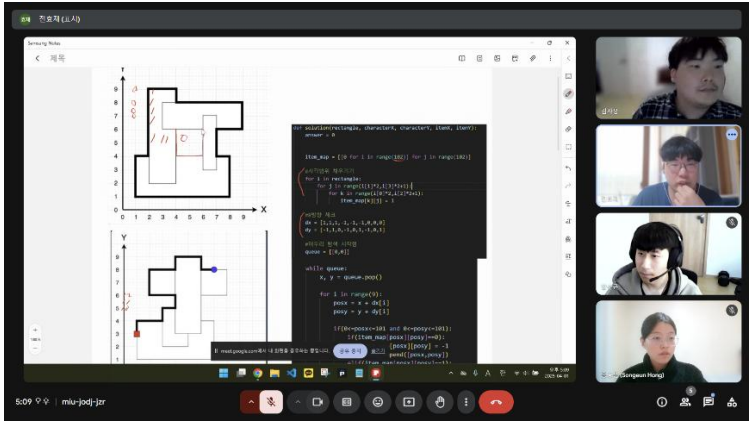
- 테두리를 구하기 위해 사각형 내부에서가 아닌 외부에서 DFS탐색을 시작
- 코너부분을 탐색하지 못하기 때문에 4방향(상하좌우)이 아닌 8방향으로 탐색
- ㄷ자 모양에서 길이 아닌 방향도 길로 인식하는 오류가 있기 때문에, 경로 탐색 범위를 기존의 가로와 세로에서 두 배로 확장하여 중앙 영역까지 포함하도록 처리
- 테두리 시작점에서 종점까지 두 방향 거리 중 최소값 리턴

- 여행 경로 찾기(김사웅)

: 주어진 시작지점과 도착지점 쌍의 배열을 활용해 문자열 순으로 빠른 경로를 찾는 문제

- 인접 배열로 문자 데이터를 변경
- 재귀 함수를 활용해 DFS 구현으로 갈 수 있는 깊은 노드까지 탐색한다. (최단 거리가 아니어도 됨)
- 문자열 순으로 빠른 문자를 리턴하도록 함
- 팝하는 구조 고려하여 리버스로 문자열 정렬을 시행

	<ul style="list-style-type: none"> • 네트워크(장수근) <p>: 직접적으로 연결된 컴퓨터의 개수와 연결상태를 입력받아 네트워크의 개수 출력</p> <ul style="list-style-type: none"> ◦ 연결이 끊어진다면 네트워크가 분리된것으로 판단할 수 있기 때문에 끝까지 탐색하는 깊이 우선 탐색 사용 ◦ 무방향 그래프의 대칭 구조라서 대각선은 본인이라 탐색 조건에서 제외 ◦ 더이상 탐색 하지 못할 경우 연결이 끊어진 것으로 네트워크의 개수 + 1 ◦ 모든 노드를 탐색해야 하는 문제 특성상 너비 우선 탐색도 사용 가능 	
활동평가	전효재	DFS 개념을 학습하고 아이템 줍기 문제에 직접 적용해보며 이해도를 높임. 탐색 범위 설정과 8방향 탐색을 구현하며 DFS의 활용법을 익혔고, 재귀 방식 풀이도 고민함. 다른 사람의 코드를 참고하며 다양한 접근 방식을 비교하고, 효율성을 개선하는 방법까지 고민함.
	홍송은	BFS와 비교하여 DFS의 개념을 학습함. DFS 알고리즘 문제 - 타겟 넘버를 풀이하며 직접 트리를 그리며 DFS를 활용한 재귀 함수를 작성함. 단순히 정답 코드만 작성하는 것이 아닌, 작성한 코드의 효율성을 높이기 위해 고민하며 간결한 코드로 수정하는 과정을 설명함.
	김사웅	BFS에 나아가 DFS를 적용함으로써 둘의 차이를 인지하게 됨. 하지만 아직 문제가 나왔을 때 어떤 걸 적용해야 잘 풀 수 있을지는 더 많은 문제를 풀어봐야 한다고 생각함. 최단 거리를 구하기보다 깊은 경로로 탐색하여 필요한 조건을 대입해 푸는 방식으로 문제를 풀었음.

	장수근	DFS/BFS를 실제로 코딩하는 것은 처음이라 어려움이 많았지만 스터디를 통해 차이점과 주의사항을 숙지할 수 있었음. 비교적 난이도가 쉬워서 해당 문제를 DFS와 BFS 모두 사용하여 푸는 기회를 얻었고, 더욱 많은 문제를 풀어서 어떤 탐색 방식을 사용해야 하는지를 판별하는 것이 숙제.
과제		<ul style="list-style-type: none"> 제출 기한: 4월 4일(금) 23:59까지 <ul style="list-style-type: none"> 시험 대비용으로 20차시별 예상 문제를 각자 1문제씩 총 20문제 출제 작성한 문제는 슬랙의 구글 docs에 업로드하여 팀원들과 공유 풀이 기한: 4월 5일(토) 스터디 전까지 <ul style="list-style-type: none"> 공유된 총 80문제(팀원 4명 × 20문제)를 모두 풀어오기 풀이 내용은 별도 양식 없이 개인 정리로 진행
향후 계획		<ul style="list-style-type: none"> 4주차 스터디는 4월 9일(화) 파이썬 정기평가 대비를 위해 날짜 조정 <ul style="list-style-type: none"> 4월 4일(토) 오후 8시 30분 각 주차별 학습 내용 복습 차시별 예상 문제 작성 및 풀이를 통해 실전 감각 강화 출제된 예상 문제 중 주요 문제 풀이 및 피드백 진행
첨부 자료	스터디 화면	

	<div>결과물</div>	<div><p>Github 주소:</p><p>https://github.com/yellowHSE/ds_rokey4_study_team2/tree/main/codes/week3_DFS</p><div><div><div>Files</div><div>main</div><div>Go to file</div><div>codes</div><div>week2_BFS</div><div>week3_DFS</div><div>DFS_Hyojae.py</div><div>DFS_Saung.py</div><div>DFS_Songeun.py</div><div>DFS_Sugeun.py</div><div>README.md</div><div>reports</div><div>README.md</div></div><div><div>ds_rokey4_study_team2 / codes / week3_DFS /</div><div><div>알고리즘 스터디 기록</div><div>3주차 문제 - DFS(깊이 우선 탐색)</div><div><div>1. 타겟 넘버</div><div><div>문제 링크: 프로그래머스 - 타겟 넘버</div><div>코드 : DFS_Songeun.py</div></div><div><div>2. 여행 경로</div><div><div>문제 링크: 프로그래머스 - 여행 경로</div><div>코드 : DFS_Saung.py</div></div><div><div>3. 아이템 찾기</div><div><div>문제 링크: 프로그래머스 - 아이템 찾기</div><div>코드 : DFS_Hyojae.py</div></div><div><div>4. 네트워크</div><div><div>문제 링크: 프로그래머스 - 네트워크</div><div>코드 : DFS_Sugeun.py</div></div></div></div></div></div></div></div></div></div>
--	----------------	---