

최종보고서

멘사스터디 202055614 최성민

1. 공부한 내용:

2024학년도 2학기 멘사 스터디 활동을 하면서 목표했던 문제 유형들에 대한 공부와 문제 풀이를 달성할 수 있었다. 목표했던 유형은 다음과 같다.

- 그래프 탐색(+다익스트라, 플로이드 워셜), 다이나믹 프로그래밍, 이분 탐색, 구현, 자료구조, 백트래킹, 유니온 파인드, 비트마스킹

각 유형별로 이 알고리즘이 어떤 때에 사용되는지, 이 문제에 어떤 방식으로 사용됐는지에 초점을 두어 공부해서 암기식으로 문제를 푸는 것이 아니라 이해를 하고 적절한 알고리즘을 대입하여 푸는 연습을 하였다. 총 30문제를 풀이하였으며 난이도는 백준 기준 실버, 골드 난이도 문제를 풀었다.

2. 그래프 탐색

내가 항상 그래프 탐색이 어렵다고 느낀 이유가 반복문을 통해서 노드를 방문할지 방문해서는 안될지 직관적으로 생각하기 어려웠기 때문이었다. 또한, 어느 부분에서 너비 우선 탐색을 해야 할지, 깊이 우선 탐색을 해야 할지 결정하기 어려웠다. 그러나 유형별로 문제를 접하면서 14500번 테트로미노 문제와 같이 모든 경로를 거쳐 완전 탐색이 필요할 때에는 dfs, 7569번 토마토 문제와 같이 최적의 경로를 구할 때에는 bfs를 써야 한다는 것을 알 수 있었다. 더 나아가 1238번 파티 같이 시작 지점부터 도착 지점까지의 모든 경로 중에서 가장 최단거리를 구하는 문제 같은 경우는 다익스트라 알고리즘을 사용하여 문제를 해결할 수 있다는 점을 배웠다. 여기서 11403번 경로찾기 문제 같은 모든 경로를 구하는 문제는 플로이드 워셜 알고리즘을 이용해 풀이가 가능하다는 점을 알 수 있었다.

3. 다이나믹 프로그래밍(동적 계획법)

다이나믹 프로그래밍은 문제도 다양하고 다른 유형들과 접목되어 출제되는 문제들이 많아서 처음에 접하기 굉장히 어려웠다. 1463번 1로만들기 문제와 같이 간단한 동적 프로그래밍 문제를 풀면서 동적 계획법의 핵심 아이디어인 최적 부분 구조와 중복되는 하위 문제, 메모이제이션을 이해하는데 도움이 되었다. 1912번 연속합과 같은 문제는 Bottom-up 방식으로 localMax 부분합을 구하는 것이 키 포인트였고, 12865번 평범한 배낭 문제는 2D DP를 구현함으로써 한정된 무게 내에서 가치의 합이 최대가 되도록 할 수 있었다. 다이나믹 프로그래밍은 조금 더 자주 풀어보면서 몸에 익히는 것이 중요하다는 점을 알 수 있었다.

4. 이분 탐색

이분 탐색은 정렬된 배열이나 리스트에서 값을 빠르게 찾을 수 있는 알고리즘이다. 이분 탐색이라는 개념은 잘 알고 있었으나 어느 부분에서 사용해야 될지는 잘 몰랐었다.

1654번 랜선 자르기 문제에서도 그랬는데 처음에는 반복문을 통해서 1부터 랜선의 최대 길이까지 찾는 방식으로 구현을 했는데 시간이 초과될 뿐만 아니라 복잡도도 많이 높았다. 여기서 사용되는 것이 이분 탐색으로 일일이 탐색하는 것이 아니라 탐색 범위를 절반씩 줄여나가기 때문에 효율적이었다. 2512번 예산 문제가 제한 시간 짧았지만 이분 탐색을 사용해 예산 요청의 최대값을 빠르게 구할 수 있었다.

5. 자료구조

자료구조 같은 부분에서는 대개 정렬 문제에서 유용하게 사용할 수 있었다. 특히, 최소힙을 이용해서 정렬 혹은 최소값을 구하는 문제를 손쉽게 해결할 수 있었다. 1655번 가운데를 말해요 문제는 시간 제한이 0.1초로 굉장히 짧지만 heapRight는 최소힙, heapLeft는 최대힙으로 만들어 heapLeft의 root노드의 값이 중간값으로 만들어 지게끔 구현하여 해결할 수 있었다.

6. 백트래킹

백트래킹은 내게는 익숙하지 않은 문제였다. 모든 경우의 수를 탐색하면서도 불필요한 탐색을 가지치기하여 효율적으로 문제를 해결하는 알고리즘 설계 기법으로 재귀를 기반으로 최적의 해를 찾거나 가능한 해를 모두 탐색할 때 사용된다. 1759번 암호 만들기 문제가 백트래킹으로 해결하였는데 암호를 만드는 조건이 있고, 이 조건을 만족하는 가능성 있는 모든 암호를 구하는 문제이다. 백트래킹을 통해서 문자로 암호를 만들고 동시에 valid할 시 return하는 방식으로 구현하여 해결할 수 있었다.

7. 구현

구현 문제는 다른 알고리즘을 사용하기 보다는 말그대로 직접 문제를 구현하여서 해결하는 방식이다. 17144번 미세먼지 안녕! 문제는 미세먼지가 공기청정기의 바람에 따라 변하는 것을 구현해야 하는 문제이다. 미세먼지의 움직임은 기본적으로 bfs를 이용하여 다음 번에 미세먼지가 어떻게 변하는지 구현하였다. 다만, 그 변화를 바로 원래 배열에 적용시키면 값이 변해버리기 때문에 1시간이 지난 후에 변화량을 한 번에 업데이트하는 방식을 사용했다. 미세먼지가 확산되는 방식이 시계, 반시계 방향 두 가지 이므로 이를 고려해서 정밀하게 구현하는 것이 어려웠던 점이였다.

8. 유니온 파인드

유니온 파인드는 서로소 집합을 표현하고 효율적으로 관리하기 위한 자료구조로, 주로

그래프 알고리즘에서 사이클 검출, 최소 신장 트리 생성 등에 사용 된다고 한다. 여기서 Find는 특정 요소가 속한 집합의 대표(루트) 요소를 찾는 연산이고, Union은 두 집합을 하나로 합치는 연산이다. 이 부분에서 어느 한 집합의 부모 집합을 찾을 때에는 유니온 파인드를 이용하면 손쉽게 해결할 수 있음을 알 수 있었다. 2606번 바이러스 문제는 컴퓨터가 웜 바이러스에 걸리면 그 컴퓨터에 연결된 네트워크 상의 모든 컴퓨터에게 옮기 게 되는데 1번 컴퓨터가 바이러스에 걸리면 바이러스에 옮는 컴퓨터가 몇 개인지 구하는 문제이다. 여기서 네트워크를 구현하는 부분에서 Union방식으로 집합을 하나로 합치고, Find를 통해 부모가 1번인 컴퓨터의 개수를 구하여서 문제를 해결할 수 있었다.

9. 비트마스킹

비트마스킹은 문제 유형 중에서도 생소하고 어려웠던 부분인데 주로 집합을 표현하거나 특정 조건을 확인 및 변경할 때 사용되며, 비트 연산을 활용해 높은 성능을 제공한다는 이점이 있다. 9527번 1의 개수 세기 문제는 시간 제한이 1초로 굉장히 짧다. A, B 두 자연수가 주어지고 둘 사이의 자연수를 이진수로 바꿨을 때 1의 개수를 모두 세는 문제로 일일이 더해서 1의 개수를 세면 시간초과가 된다. 이 때 비트마스킹 방식을 사용하여 2의 60제곱까지의 누적합을 구해놓고 B까지의 1의 누적합과 A까지의 1의 누적합을 빼는 식으로 구현하면 문제를 해결할 수 있다.

10. 결론 및 느낀점

스터디를 통해서 나에게 특히 어려웠던 그래프 탐색 문제를 잘 해결할 수 있었고, 유니온 파인드와 비트마스킹, 백트래킹 등 익숙하지 않았던 유형들도 공부할 수 있는 기회가 되었다. 예전에는 문제를 보면 이해하고 방식을 찾는 것이 아니라 생각 나는대로 문제를 풀어서 효율적이지 못하고 정답률도 낮았는데, 문제의 유형을 찾고 그 유형에 맞는 알고리즘을 대입하는 연습을 통해서 정답률을 높일 수 있었다. 아직 백트래킹과 비트 마스킹과 같이 익숙하지 않은 부분은 부족하여서 문제를 더 많이 접해보아야 겠다고 느꼈고, 동적 계획법과 그래프 탐색 문제는 꾸준히 문제를 풀어보며 더 자연스러워지게끔 노력해야 겠다고 생각이 든다. 이번 활동을 통해 알고리즘에 대한 접근성을 높일 수 있어 유익했고 이 활동을 통해 배운 점을 잊지 말고 계속 공부하기로 다짐했다.