

a. Describe any difficulties you experienced while doing this assignment.

과제가 전반적으로 많이 생각하도록 하고 힘들었지만, 가장 힘들었던 부분은 Branch and Bound였습니다. 그 이유는 제가 강의 슬라이드와, 구글링 등을 통해서 여러 코드를 완성하고 돌려보았을 때, Branch and Bound 케이스의 알고리즘이 가장 오랜 시간이 걸렸기 때문입니다. Item의 갯수가 10000개인 경우에는 결과값이 20분이 지나도록 나오지 않았었기 때문에 해결방법을 찾아야 했습니다. 그래서 그 해결방법을 찾는 과정이 가장 힘들지 않았나 싶습니다. 그래도 queue를 사용하여 효율적으로 코드를 만들 수 있었습니다.

b. What are the notable differences between your code (at least the one you had in you mind) and the one you referenced on the internet?

인터넷에 존재하는 여러 코드들과는 다르게 시드를 설정하고, 'mt19937'을 통해서 아이템을 랜덤하게 생성합니다. 저는 이 점이 인터넷의 코드들과는 다른 점이라고 생각합니다. 또한 chrono를 사용하여, 시간을 세밀하게 측정하는 점도 차이점을 가진다고 생각합니다. 마지막으로 Branch and Bound에서 queue를 사용하여 노드의 우선순위를 관리하고 효율성을 더 높이는 방법 또한 차별적인 방법이라고 생각합니다. 인터넷에서 찾아봤던 branch and bound 관련 코드들은 거의 대부분 이러한 방법을 사용하지 않는데, 그래서 10000과 같은 값이 들어갔을 때, 굉장히 오랜 시간이 걸리게 됩니다.

c. Describe what you had learned from this homework.

각각의 접근법들에 대해서 더 자세하게 배울 수 있었습니다. 말로만 듣고, 생각만 해보는 것이 아니라 직접 코딩을 하고, 그 결과를 지켜보면서 더 크게 체감할 수 있었습니다. Brute Force를 통해서 시간 복잡도와 공간 복잡도라는 것을 더 잘 알 수 있었습니다. 중간고사를 풀 때, 시간 복잡도는 이해를 했지만, 공간 복잡도라는 것은 이해를 잘 하지 못했었는데, 추상적으로나마 이해할 수 있었습니다. Greedy는 최선의 선택을 하기 때문에 다른 코드들에 비해서는 더 빠른 속도를 가졌다는 것을 결과를 보면서 직접 체감할 수 있었습니다. Dynamic Programming을 통해서 최적의 해를 찾는 과정을 코딩하는 법에 대해서 생각해 볼 수 있었습니다. Branch and Bound를 통해서 상한과 하한을 정해두는 법을 이해할 수 있었습니다. 그리고 처음에는 BB의 경우가 가장 오래 걸리고 잘 돌아가지 않았었는데, 이를 해결하기 위해서 지속적으로 여러 방법을 찾고, 탐구하면서 코드가 돌아가는 시간, time complexity를 줄이는 방법에는 여러가지가 있었다는 것을 알 수 있었습니다. 전체적으로 여러 알고리즘을 비교해보면서 각각의 방식과 장점 혹은 단점을 더 자세하게 알 수 있었고, 해당 알고리즘들을 개선하기 위해서 여러 방법들을 대입해보면서 코딩 실력을 기를 수 있었습니다.