Homework-2 Report

I made use of java's class structure (object orainted programming) to make the project more readable and more useful. I opened a class called Heros and kept the characteristics of the heroes (gold, attack power, type) in this class. Thus, I accessed every data I needed more easily**.**

Random Approach:

In the random approach, I determined the level and index randomly and tried to get the heroes from that level and index. But I encountered some problems in the random approach. For example, I encountered the error of choosing more than one hero from the same level. Before that, I kept the random numbers in an arraylist. If the random number is in the arraylist, I encountered another random number. I got the numbers drawn.

Greedy Approach:

In Greedy's algorithm, I did it using nested loops. With two nested for loops, I wandered through the allowed number of pieces at five levels. And I took the hero with the most attack power. Here I encountered the same problems that I encountered in the random approach and kept an integer number as a solution. A number that formulates that it's in the index. Using this, I selected only one hero with the highest attack power as long as we had enough gold at each level without any problems.

Dynamic Approach:

In dynamic programming, I tried to adapt the knapsack algorithm to my code. But since I kept an array of Heros type, it was not very suitable for the knapsack algorithm. For this, I created a Heros type array called herosArrayDynamic and threw the heroes in the allowed level and index into it. I could not access. I printed the total attack points.

References:

https://www.geeksforgeeks.org/java-program-for-dynamic-programming-set-10-0-1-knapsack-problem/