



DOKUZ EYLUL UNIVERSITY
ENGINEERING FACULTY
DEPARTMENT OF COMPUTER ENGINEERING



CME 2204 ALGORITHM ANALYSIS
ASSIGNMENT 1 REPORT

by
Muzaffer Sevil
2019510069

Lecturers
Assoc. Prof. Dr. Zerrin IŞIK
Res.Asst. Ali CUVİTOĞLU
Res.Asst. İbrahim BERBER

İZMİR
22.05.2022

CONTENTS

1.INTRODUCTION

1.1 Random Approach

1.2 Greedy Algorithm Approach

1.3 Dynamic Programming Approach

2.PERFORMANCE CALCULATIONS

3.ALGORITHM OF SCENARIO

4.REFERENCES

1.INTRODUCTION

The aim of the project is to create an army with chess pieces that have historical names and different attack points using different approaches. Then, the analysis of the approaches is done according to the data results.

Three approaches were used: Random Approach, Greedy Algorithm Approach, and Dynamic Programming Approach.

1.1 Random Approach

- The setStatus method is called to reset the selections.
- Pieces are randomly selected according to the maximum_allowed_level, but computer do not have to choose from every level. For example, if maximum_allowed_level is 7, but there is enough money for the pieces in only the first 4 selections, the army will be created according to only the first 4 selections.
- The selection order is random. For example, the pawn does not have to be chosen first, the king can also be selected randomly.
- The setSelected function is called in a for loop from the Hero class where the heroes are kept to select only one item of the same level.

setStatus(): $O(n)$ time

nested for loop= $O(n*n)$ time

$T(n) = O(n) + O(n*n)$

$T(n) = O(n*n)$

1.2 Greedy Approach

- The setStatus method is called to reset the selections.
- Pieces are selected based on the highest attack points according to the maximum_allowed_level, but the computer does not have to choose from every level. For example, if the maximum_allowed_level is 8 and the first 5 selections have maximum attack points for the pieces, only the first 5 selections will create an army.
- The selection order is based on the highest attack point/cost ratio. For example, if the piece with the highest attack points/cost ratio is one of the bishops, it will be the first choice.
- The setSelected function is called in a for loop from the Hero class where the heroes are kept to select only one item of the same level.

setStatus(): $O(n)$ time
nested for loop= $O(n*n)$ time
 $T(n) = O(n) + O(n*n)$
 $T(n) = O(n*n)$

1.3 Dynamic Programming Approach

- The setStatus method is called to reset the selections.
- Pieces are selected based on the highest attack points/cost ratio according to the maximum_allowed_level, but the computer does not have to choose from every level. For example, if the maximum_allowed_level is 5 and only the first 3 selections have enough money for the pieces, only the first 3 selections will create an army.
- The established army is the army with the maximum attack points that can be established with the money in hand.

setStatus(): $O(n)$ time
for loop= $O(n)$ time
nested for loop= $O(n*n)$ time
 $T(n) = O(n) + O(n) + O(n*n)$
 $T(n) = O(n*n)$

2.PERFORMANCE CALCULATIONS

GOLD=1200 MAX_LEVEL=9 LEVEL_PER_PIECE=10

- Greedy ~=0 seconds
- Random ~=0 seconds
- Dynamic ~= 5 seconds

GOLD=1200 MAX_LEVEL=5 LEVEL_PER_PIECE=4

- Greedy ~=0 seconds
- Random ~=1 seconds
- Dynamic ~= 1 seconds

GOLD=600 MAX_LEVEL=9 LEVEL_PER_PIECE=10

- Greedy ~=0 seconds
- Random ~=1 seconds
- Dynamic ~= 3 seconds

GOLD=600 MAX_LEVEL=7 LEVEL_PER_PIECE=8

- Greedy ~=0 seconds
- Random ~=0 seconds
- Dynamic ~= 0 seconds

3. ALGORITHM SCENARIO

- In Dynamic Programming Approach I used Knapsack Algorithm for implementation. But I could not implement cost calculation and printing selected heroes in this approach.
- In Random Approach, selections were randomly with Random built-in method.
- In Greedy Approach, selections made with best attack point/cost ratio.

4. REFERENCES

- Lessons