Coin Change Problem using Brute Force and Greedy Algorithm

Chamirti Senthilkumar (RA2111003010504)

Manasa Pokala (RA2111003010541)

Objective

To determine the efficiency and accuracy of the two mentioned algorithms in finding the currency exchange values. This can be measured by analyzing the time complexity of the algorithms. Additionally, other factors such as the ease of implementation, the ability to handle larger amounts are also taken into account.

Abstract

Automatic money exchange systems are very rarely found but are needed at certain times. The purpose of this project is to make it easier for people who want to exchange money from big to small nominal. The obstacle faced in this exchange system is the use of algorithms needed to run the system. Some method that can be used to design and implement this system is using Greedy Algorithm and Brute force algorithms. By using this greedy algorithm, a problem with the shortest route search technique can be completed quickly, but the results generated by the greedy algorithm are not always optimal, while brute force can solve the problem of money exchange optimally but need a longer time. In this subject we trying to solve this problem by making an application using python language to implement the greedy and brute force algorithm in solving of this problem. After making the code algorithm and implementing the greedy and brute force method, we conclude that our program can solve this coins exchange problem just in a second but sometimes the result is not optimal (with the most smallest fraction), but if using brute force algorithm we can get the optimal total of money changes but the time will extremely increase up to 20 times higher depend on the amount of the total money that want to be changes. in this experiment the user will input the money that want to change, in the program we already have a predetermined amount and money that has been prepared to exchange money

Introduction

-***Background of the problem***

In daily life we always deal with financial problems, one of the problem which is how the cashier gives change money. This report will discuss the comparison of Greedy and Brute Force algorithm in the example of optimization problem, namely the problem of money exchange. In this case we want to see and know which algorithm has an optimum result and is faster. In this experiment we will exchange money in rupiah currency. In this case we will Exchange money with a predetermined amount and money that has been prepared to exchange money. The money will be exchanged using smaller money and will give the minimum and equivalent to the money that has been exchanged. The method used to solve this coin-change problem is Brute Force and Greedy Algorithm which will be used for comparison

- Brief Problem

In this paper we will analyze two algorithms which one is better between Brute force and Greedy algorithm to solve Coin change problem, and we will use Python as the programming language to solve the problem.

Analysis of Algorithm

Greedy Algorithm

The Greedy algorithm is the algorithm most often used in solving optimization problems. Greedy itself is taken from English, which means “serakah”, “tamak” or “rakus”. In accordance with these meanings, the Greedy principle is "Take what you can get now". The greedy algorithm will form a step-by-step solution. At each step, we must make choices that can produce optimum results. Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. So the problems where choosing locally optimal also leads to global solution are best fit for Greedy.

Brute Force Algorithm

Brute Force Algorithms refers to a programming style that does not include any shortcuts to improve performance, but instead relies on sheer computing power to try all possibilities until the solution to a problem is found. The Brute force algorithm is a straightforward approach to solving a problem, usually based on a problem statement. The brute force algorithm solves problems very simply, directly and in a clear way (obvious way)

Time Complexity Analysis of Algorithms

Since in our program we use Brute Force and Greedy Algorithm to find the optimum and minimal money change so, the time complexity are

a) Brute force

Number of Coins in the array: 4

The number of coins to be exchanged: n

Time Complexity: O(4 ^ n)

b) Greedy Algorithm

Number of Coins in the array: 4

The number of coins to be exchanged: n

Time complexity: O(N)

Study case

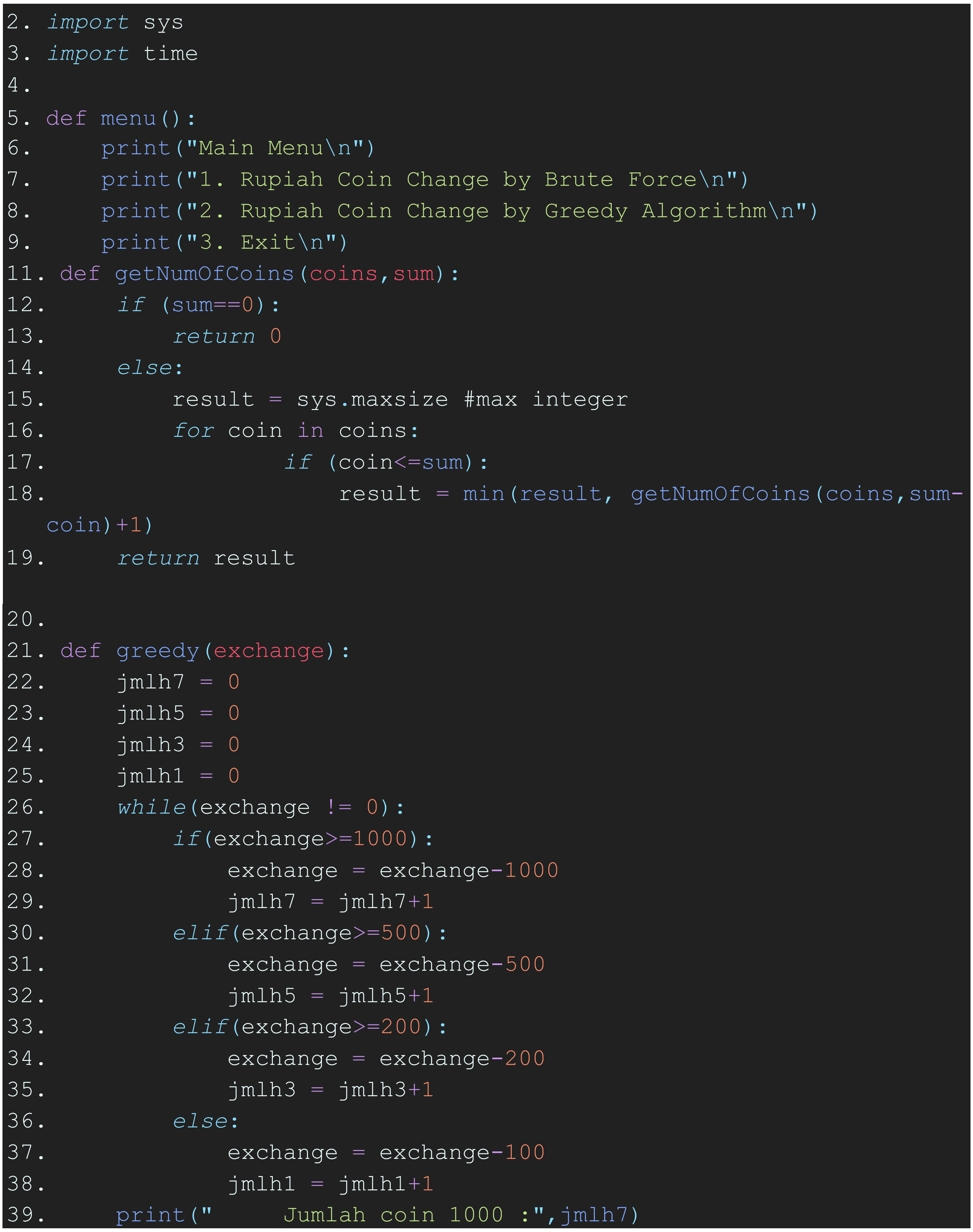
The program has provided Rupiah in the form of fractions [100, 200, 500, 1000] and will receive input money that will be exchanged by the user. Users can see the minimum output of money that has been exchanged and the execution of the time needed by the program.

Computer specifications

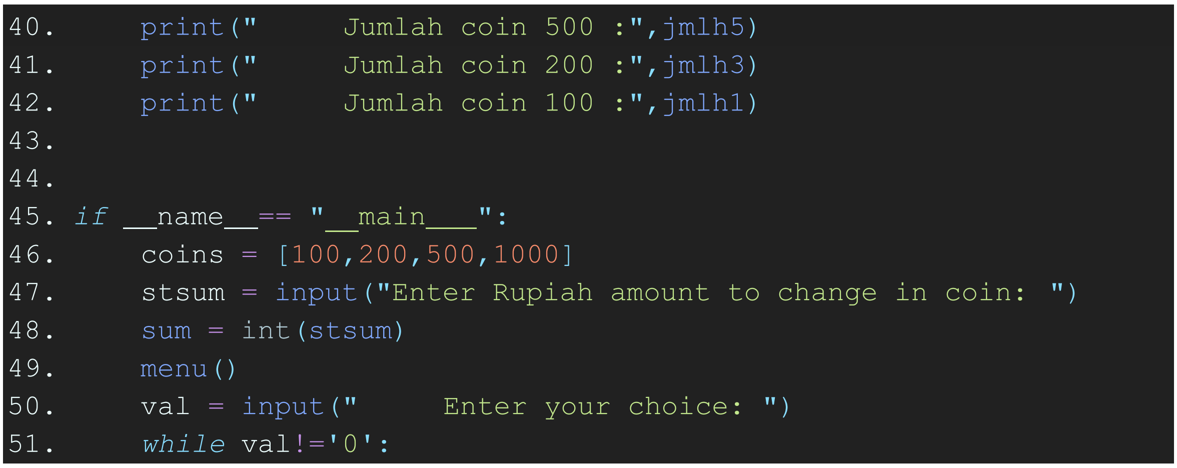
Processor	Intel Core i7-4750HQ
Memory	12GB DDR3L
Hard Drive	240 GB SSD (OS Drive) + 1 TB SSD
Operating System	Windows 10

Experimental Results

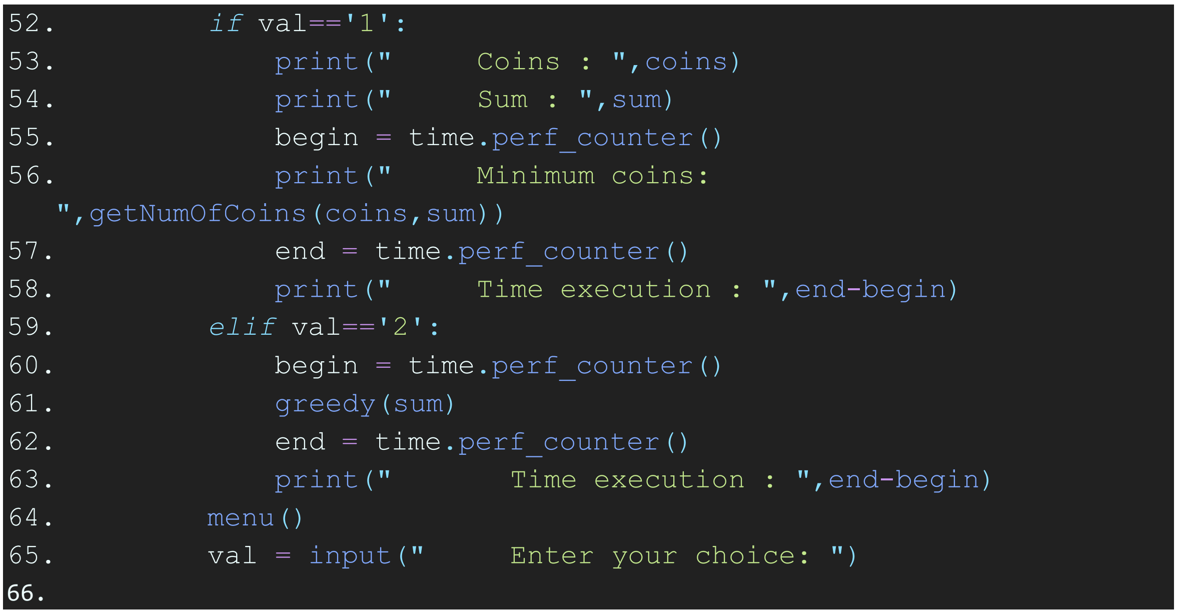
Program code



```python
import sys
import time

def menu():
    print("Main Menu\n")
    print("1. Rupiah Coin Change by Brute Force\n")
    print("2. Rupiah Coin Change by Greedy Algorithm\n")
    print("3. Exit\n")
def getNumOfCoins(coins,sum):
    if (sum==0):
        return 0
    else:
        result = sys.maxsize #max integer
        for coin in coins:
                if (coin<=sum):
                    result = min(result, getNumOfCoins(coins,sum-coin)+1)
    return result


def greedy(exchange):
    jmlh7 = 0
    jmlh5 = 0
    jmlh3 = 0
    jmlh1 = 0
    while(exchange != 0):
        if(exchange>=1000):
            exchange = exchange-1000
            jmlh7 = jmlh7+1
        elif(exchange>=500):
            exchange = exchange-500
            jmlh5 = jmlh5+1
        elif(exchange>=200):
            exchange = exchange-200
            jmlh3 = jmlh3+1
        else:
            exchange = exchange-100
            jmlh1 = jmlh1+1
    print("     Jumlah coin 1000 :",jmlh7)
```

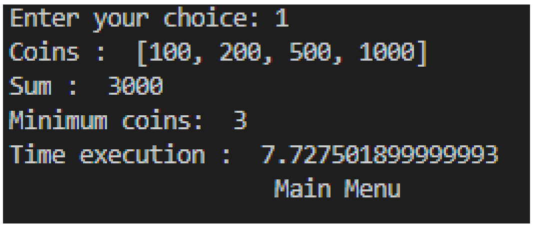


```python
    print("     Jumlah coin 500 :",jmlh5)
    print("     Jumlah coin 200 :",jmlh3)
    print("     Jumlah coin 100 :",jmlh1)


if __name__== "__main___":
    coins = [100,200,500,1000]
    stsum = input("Enter Rupiah amount to change in coin:  ")
    sum = int(stsum)
    menu()
    val = input("     Enter your choice: ")
    while val!='0':
```



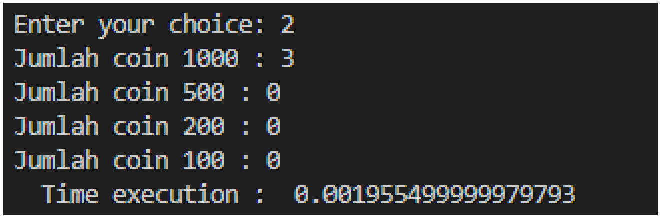
```
        if val=='1':
            print("     Coins : ",coins)
            print("     Sum : ",sum)
            begin = time.perf_counter()
            print("     Minimum coins: ",getNumOfCoins(coins,sum))
            end = time.perf_counter()
            print("     Time execution : ",end-begin)
        elif val=='2':
            begin = time.perf_counter()
            greedy(sum)
            end = time.perf_counter()
            print("       Time execution : ",end-begin)
        menu()
        val = input("     Enter your choice: ")

```

a) The Output from Brute force

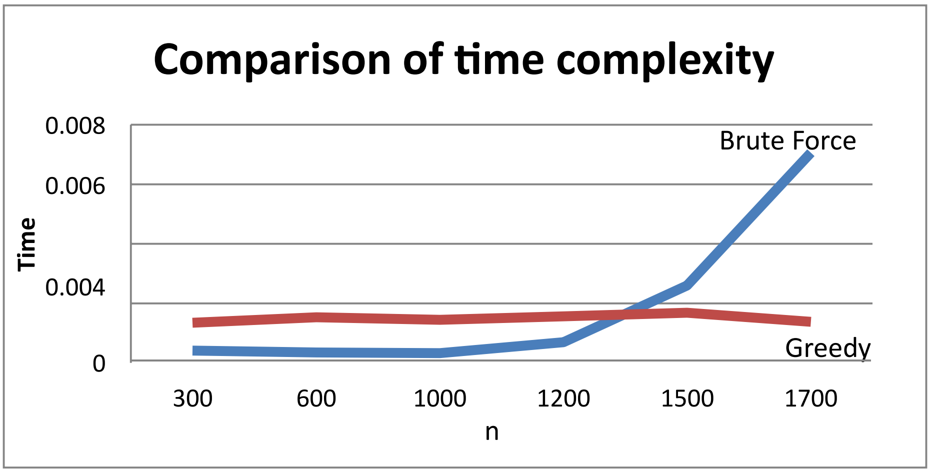


```
Enter your choice: 1
Coins :  [100, 200, 500, 1000]
Sum :  3000
Minimum coins:  3
Time execution :  7.727501899999993
                   Main Menu
```

b) Greedy Algorithm



```
Enter your choice: 2
Jumlah coin 1000 : 3
Jumlah coin 500 : 0
Jumlah coin 200 : 0
Jumlah coin 100 : 0
  Time execution :  0.0019554999999979793
```



Graph 1 Comparison of time complexity from two algorithms.

From graph 1, the greedy algorithm (red) and brute force (blue) have different time executions. as in the example of exchanging money for Rp.1,700, for bruteforce time in problem solving is greedy algorithm almost 7x faster to solve the coin change problem when greedy is done. From this experiment it can be concluded that the greedy algorithm is faster even though the results are not always optimum.

For Brute force the more money will be exchanged, the longer the execution time will be, while the Greedy Algorithm has an execution time that is faster than brute force.

Conclusion

From the comparison of the Greedy algorithm and brute force in our experiments, it can be concluded that the greedy algorithm can produce results that are faster than brute force, in example on the total coin change 1700 rupiah, greedy algorithm almost 7x faster to solve the coin change problem when greedy done to execute the program in 0.0013787 second and bruteforce done in 0.0070634 second. However, the results of the money exchange from the greedy algorithm cannot always produce optimalresults.

References

[1] Anindhika, Dita. "Komplekstas Algoritma untuk Penyelesaian Persoalan Penukaran Koin dengan Algoritma Greedy", Program Studi Teknik Informatika, Institut Teknologi Bandung.

[2] Logianto, Albert. “Penyelesaian Persoalan Penukaran Uang dengan Program Dinamis”. Program Studi Teknik Elektro dan Informatika, Institut Teknologi Bandung.

[3] https://dosenit.com/ilmu-komputer/pengertian-algoritma-brute-force-dan-greedy