Marktwage

Code:

*Programmiersprache: Python*

*Auf Github, unter* *<https://github.com/AndreasMuehlmann/Marktwage>. Branch: main*

Beschreibung:

Class Node:

* Count: Wie oft hintereinander wird ein einzelnes Gewicht verwendet
* Weight: Das letzte Gewicht, das dazu gekommen ist
* Total\_weight: Das insgesamte Gewicht
* Previous: Der Index des vorherigen Knoten
* Index: Der Index dieses Knotens

Funktion \_\_post\_init\_\_:

* Das gesamte Gewicht wird aus dem vorherigen gesamten Gewicht, der Anzahl des letzten Gewichts und dem letzten Gewicht berechnet

Funktion main:

* Die Gewichte werden von der Website gelesen (Funktion get\_weights)
* Alle möglichen Kombinationen werden probiert und als Knoten gespeichert (Funktion get\_combinations)
* Der Knoten mit dem Gewicht das am nächsten zum gesuchten Gewicht ist wird gegeben (Funktion get\_closesed\_weight)
* Ein Weg wird durch die gespeicherten Nodes zu dem nächsten Gewicht gefunden (Funktion get\_path)
* Dieser wird ausgegeben (Funktion print\_path)
* Die Zeit die das Programm gebraucht wird ausgegeben
* Die Zeit die find\_combination beansprucht hat wird ausgegeben

Funktion get\_weights:

* Die Website wird von der requests library angesprochen und es wird eine Antwort gegeben
* Der Inhalt dieser Antwort wird nach dem „utf-8“ decodiert
* Dieser wird formatiert zurückgegeben

Funktion get\_combinations:

Die einzelnen Lagen des Baumes werden durchgegangen und es wird ein Gewicht mit all seinen Möglichkeiten angehängt, d.h., dass wenn man ein Gewicht öfter als einmal hat werden die Möglichkeiten 1\*Gewicht, 2\*Gewicht, … angehängt. Weil man bei einer Marktwage ein Gewicht auf beide Seiten stellen kann, kann man nicht nur zu dem Insgesamten Gewicht addieren, sondern auch davon subtrahieren.

Also gibt es die Möglichkeiten: 1\*Gewicht, 1\*-Gewicht, 2\*Gewicht, 2\*-Gewicht, …

Das funktioniert, da Gewicht – Gewicht = 0, also sind solche Möglichkeiten vernachlässigbar. Bei jeder Lage wird ein Knoten mit dem Gewicht 0 hinzugefügt, da bei jeder einzelnen Lage jeweils die Möglichkeiten eines Gewichts angehängt werden und dadurch kann man die Möglichkeit Gewicht\*0 vernachlässigen. Jeder Knoten der besucht wird, wird gespeichert. Schlussendlich muss noch die letzte Lage zu den besuchten Knoten hinzugefügt werden, da diese nicht mehr besucht werden muss, weil nichts mehr angehängt werden muss. Dann werden die besuchten Gewichte zurückgegeben.

Funktion Get\_closesed\_weight:

* Alle Knoten werden nach dem nächsten Gewicht durchsucht
* Wenn ein Gewicht näher an dem gesuchten ist als die zuvor wird es gespeichert
* Der Knoten mit dem nächsten Gewicht wird zurückgegeben

Funktion Get\_path:

* Dem Weg wird der Übergebene Knoten hinzugefügt
* Der vorherige Knoten des übergebenen Knoten wird Get\_path übergeben (rekursiv)
* Bis es keinen vorherigen Knoten gibt
* Der ganze Weg wird zurückgeben

Funktion print\_path\_for\_weight:

* Das gesuchte Gewicht wird ausgegeben
* Dann die einzelnen Schritte des Weges
* Dabei wird die Anzahl einzelner Gewichte beachtet
* Das erreichte Gewicht wird ausgegeben