Pex-Max  Pex-Max  Inhaltsverzeichnis  Lösungsidee  1. Aufbau  2. Umsetzung  3. Beispiele  4. Quellcode  Lösungsidee  Under Stelle der Zahl, alle 15+1 Möglichkeiten sie zu verändern anzuschauen, und dann einen Depth-First-Search Algorithmus daruber laufen zu lassen. Es wird immer mitgezählt, wieviele Segmente genommen/platziert werden, und nur Veränderungen, die das Maximum minicht überschreiten
kommen infrage. Am Ende der Zahl/des Displays (Index == len (display) wird gecheckt, ob die Zahl der genommenen und platzierten Stäbchen übereinstimmt, ansonsten wird eine weitere Möglichkeit zurückverfolgt.
Aufbau  segment.py  class Segment  Klasse, die ein Segment einer 7-Segment-Anzeige repräsentiert  definit(char: str)  Initialisiert das Segment von einem Zeichen [0-9A-F]  defrepr()  Gibt das Segment in lesbarer Form aus  defeq(other)  Gibt als Wahrheitswert zurück, ob das Segment identisch zum Segment other ist  def ascii_art() -> List[str]  Produziert ascii-art um das gegebene Segment auf 3x3 Zeichen anzuzeigen
def get_takes_gives(seg) -> Tuple[int, int]  program.py  def get_max_swappable(segments: List[Segment], m: int) -> str  Gibt die Maximalzahl mit m Umlegungen zurück  def_animate(from_: List[Segment], to: List[Segment]) -> Generator[List[Segment], None, None]  Animiert die Umlegungen vom Display from_ zum Display to.  def_print_asciiart(display: List[Segment])  Gibt das Display display als ascii-art in die Konsole aus  Umsetzung  Das Programm ist in der Sprache Python umgesetzt. Der Aufgabenordner enthält neben dieser Dokumentation eine ausführbare Python-Datei program.py. Diese Datei ist mit einer Python-Umgebung ab der Version 3.6 ausführbare.
Wird das Programm gestartet, wird zuerst eine Eingabe in Form einer einstelligen Zahl erwartet, um ein bestimmtes Beispiel auszuwählen. (Das heißt: 0 für Beispiel hexmax0.txt)  Nun wird die Logik des Programms angewandt und die Ausgabe erscheint in der Kommandozeile.  Beispiele  Hier wird das Programm auf die sechs Beispiele aus dem Git-Repo angewendet:  hexmax0.txt  D24 3  Umlegungen
Ausgabe zu hexmax0.txt  EE4  hexmax1.txt
### ##################################
**************************************
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
hexmax4.txt  1A02B6B50D7489D7708A678593036FA265F2925B21C28B4724DD822038E3B4804192322F230AB7AF7BDA0A61BA7D4AD8F888 87  Ausgabe zu hexmax4.txt  FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
<pre>Quellcode segmentsy  from typing import Callable, Generator, List, Mapping, Tuple, Union  class Segment:     ""Class representing a segment of a 7-segment display."""  definit(self, chan: Union[str, Tuple[Union[int, bool]]]):     Initialise the segment with data.  Parameters    </pre>
List(str)  A 33 matrix of characters traversing every row from the top left to the bottom right.    0   1   2   \n
<pre>from itertools import poin, dirmane from typing import List, union, Tuple, Generator from segment import Segment  costump. List[List[Tuple[int, int]]] - []  # create costump O(1) for x, from, in enumerate("01236597808CDSF"):</pre>
<pre>out = [[], [], []] for seg in display:     asciiart = seg.ascii_art()     for i in range(3):         out[i] += asciiart[i*3:i*3+3] for line in out:     print(''.join(line))  while True:     try:     choice = int(input("Bitte die Nummer des Beispiels eingeben [0-5]: "))     with open(join(dirname(_file), f'beispieldaten/hexmax{choice}.txt')) as f:         display = [Segment(char) for char in f.readline().strip()]     m = int(f.readline().strip())      print(get_max_swappable(display, m))     except Exception as e:         print(e)</pre>