

## Stick Lengths feladatmegoldás

A kód célja, hogy meghatározza, mekkora a minimális költsége annak, hogy egy adott számú, különböző hosszúságú botot egyenlő hosszúságúvá tegyünk. Az egyes botok hosszának módosítása költséget jelent, ami a hosszkülönbségek abszolút értékével egyenlő. A medián (A medián egy statisztikai mérőszám, amely egy adathalmaz középső értékét jelenti, ha az adatokat növekvő sorrendbe rendezzük. A medián tehát egy olyan érték, amely elválasztja az adatok alsó és felső felét.) használata garantálja, hogy az abszolút különbségek összegét minimalizáljuk. Függvény definíciója:

```
1 def minimalis_koltseg_azonos_botokhoz(botok_szama, botok):
```

Függvény neve: `minimalis_koltseg_azonos_botokhoz` – leírja, hogy a botok egyenlővé tételéhez szükséges minimális költséget számítja ki; Paraméterek: `botok_szama`: a botok száma; `botok`: egy lista, amely tartalmazza a botok hosszát. Botok rendezése:

```
2     # Botok rendezése
3     botok.sort()
```

Miért rendezzük a botokat? A medián számításához szükséges, hogy a lista rendezett legyen. A rendezett lista biztosítja, hogy a medián helyes eredményt adjon.

Példa: Ha `botok = [5, 1, 3]`, akkor a rendezés után: `botok = [1, 3, 5]`.

Medián kiszámítása:

```
5     # Medián kiszámítása
6     median = botok[botok_szama // 2]
```

`botok_szama // 2`: a lista középső indexét határozza meg. Medián: a rendezett lista közepén álló érték, amely minimalizálja az abszolút különbségek összegét. Hogyan működik a medián? A medián használata garantálja, hogy a legkevesebb költség mellett érjük el az összes bot hosszának azonos értékre hozását. Példa 1 (páratlan elemszám): `botok = [1, 3, 5]`; Középső index:  $3//2=1$ ;  $3//2=1$ ; Medián: `botok[1] = 3`. Példa 2 (páros elemszám): `botok = [1, 3, 5, 7]`; Középső index:  $4//2=2$ ;  $4//2=2$ ; Medián: `botok[2] = 5`. Költség kiszámítása:

```
8     # Költség számítása
9     koltseg = sum(abs(bot - median) for bot in botok)
```

A `for bot in botok` végigmegy a lista minden elemén; `abs(bot - median)`: egy bot hosszának különbsége a mediántól (abszolút értékben, mivel a negatív

különbség is költség); `sum(...)`: az összes abszolút különbséget összeadja, ami a teljes költséget adja. Példa: ha `botok = [1, 3, 5]` és `median = 3`: Első bot:  $|1-3|=2$  |1 - 3| = 2 |1-3|=2; Második bot:  $|3-3|=0$  |3 - 3| = 0 |3-3|=0; Harmadik bot:  $|5-3|=2$  |5 - 3| = 2 |5-3|=2; Teljes költség:  $2+0+2=4$  2 + 0 + 2 = 4. Költség visszaadása:

```
11     return koltseg
```

A függvény visszaadja a számított minimális költséget, amely az összes bot egyenlővé tételéhez szükséges. Bemenet kezelése:

```
13     # Bemenet
14     botok_szama = int(input())
15     botok = list(map(int, input().split()))
16
```

`botok_szama = int(input())`: bekéri a felhasználótól a botok számát. Példa: Ha a felhasználó beírja: 3, akkor `botok_szama = 3`; `botok = list(map(int, input().split()))`: bekéri a botok hosszát egy sorban, szóközzel elválasztva. A `map(int, ...)` minden számot egész számmá alakít. A `list(...)` listává alakítja az eredményt. Példa: Ha a felhasználó beírja: 1 3 5, akkor `botok = [1, 3, 5]`. Függvény meghívása és eredmény kiírása:

```
17     # Megoldás meghívása
18     eredmeny = minimalis_koltseg_azonos_botokhoz(botok_szama, botok)
19
20     # Kimenet
21     print(eredmeny)
```

A függvényt meghívjuk a bemeneti adatokkal. Az eredményt kiírjuk a képernyőre.