

## HASHING

### 💡 Hash Tables

- 🔍 **Čo to je?** Štruktúry, ktoré ukladajú informácie v určitom poradí ako páry klúč-hodnota
- 🔗 **Ako to implementovať?** V Pythone sú hash tables v podstate len slovníky (dictionaries)
- 💡 **Prečo na tom záleží?** Na rozdiel napr. od zoznamov, hash tables ukladajú a získavajú dáta v konštantnom čase ( $O(1)$ )

### 💡 Prečo sú hašovacie tabuľky také rýchle?

Hash tables používajú **hash funkciu**, ktorá premení klúče priamo na indexy v poli, takže môžu skočiť rovno tam, kde sú dáta uložené, namiesto toho, aby prehľadávali každý prvok.

### 💡 Môžu dva klúče hashnúť na ten istý index?

Áno. Toto sa nazýva **kolízia**. Ak dôjde ku kolízii dvoch záznamov, môžeme ju vyriešiť (napr. pomocou **reťazenia** alebo **otvoreného adresovania**).

## VYHLÁDÁVACIE ALGORITMY

### 💡 Príklady vyhľadávacích algoritmov

#### 🔗 Lineárne vyhľadávanie (pomalé)

```
function linear_search(array, target):  
  
    for each element in array:  
        if element == target:  
            return index of element  
  
    return -1 (not found)
```

#### 🔗 Binárne vyhľadávanie (rýchle)

```
function binary_search(array, target):  
    left = 0  
    right = length of array - 1  
  
    while left <= right:  
        mid = (left + right) / 2  
  
        if array[mid] == target:  
            return mid  
        else if array[mid] < target:  
            left = mid + 1  
        else:  
            right = mid - 1  
  
    return -1 (not found)
```