

Merge Sort [10 bodů]

Na přednášce jste probrali třídící algoritmus Merge Sort, neprovedli jste ale jeho implementaci. Připravil jsem pro Vás implementaci tohoto algoritmu. Nepřítel se však dostal k mému kódu a způsobil v něm dvě zásadní chyby. Pro můj testovací seznam k setřídění však program proběhl v pořádku, proto jsem si sabotáže nevšiml a kód Vám poslal s chybami.

Zde je: (pro lepší orientaci a dostupnost je Vám k dispozici i soubor mergeSpatne.py)

Program

```
def mergesort(data):
# Setridi seznam data pomoci algoritmu merge sort (iterativne)

    delka_useku = 1      # Aktualni delka slucovanzch useku
    delka_pole = len(data)

    while delka_useku < delka_pole:
# Hlavni smycka: slucujeme postupne
# delsi casti vstupniho seznamu

        # Slucovane useky budou levy_konec -> (stred - 1)
        # a stred -> (pravy_konec - 1)
        # indexy v promennych jsou "vpravo" od hranic useku
        levy_konec = 0
        stred = levy_konec + delka_useku
        pravy_konec = levy_konec + 2 * delka_useku

        while pravy_konec <= delka_pole:

            merge(data, levy_konec, stred, pravy_konec)

            levy_konec = pravy_konec
            stred = levy_konec + delka_useku
            pravy_konec = levy_konec + 2 * delka_useku

        # v dalsim pruchodu budeme slevat dvojnásobne useky
        # tedy useky vznikly slevanim v tomto pruchodu
        delka_useku *= 2

def merge(data, levy_konec, stred, pravy_konec):
# Sleje setridene casti v seznamu data.
# Cast od indexu levy_konec do (stred - 1)
```

```
# a od indexu stred do indexu (pravy_konec - 1).
# Spojene casti vrati do puvodniho seznamu

    # indexy i a j prochazeji levou a pravou slevanou cast
    i = levy_konec
    j = stred

    # Pripravime si seznam pro slity usek
    novy_usek = []

    while i < stred and j < pravy_konec:
        # Dokud nedojdou prvky ve slevanych castech,
        # pridavame do noveho useku vzdy mensi
        # z dosud nepridanych prvku.
        if data[i] < data[j]:
            novy_usek.append(data[i])
            i += 1
        else:
            novy_usek.append(data[j])
            j += 1

    # Novy slity usek zkopirujeme do puvodniho seznamu
    i = levy_konec
    k = 0
    while k < len(novy_usek):
        data[i] = novy_usek[k]
        i += 1
        k += 1

# Otestovani algoritmu na prikkladu

test = [0,8,4,12,2,10,6,14,1,9,5,13,3,11,7,15]

mergesort(test)

print(test)
```

Úkol. Nalezněte vstupy, pro které program nebude fungovat nebo vrátí špatný výsledek. Nalezněte místa v kódu kde se chyby vyskytují, popište v čem spočívají a opravte je (tedy s pokud možno minimálním zásahem upravte sabotovaný kód tak, aby fungoval).