

Algorytmy i Struktury Danych  
Egzamin/Zaliczenie 3 (14.IX.2023)

### Format rozwiązań

Rozwiązanie zadania musi się składać z **krótkiego** opisu algorytmu (wraz z uzasadnieniem poprawności) oraz jego implementacji. Zarówno opis algorytmu jak i implementacja powinny się znajdować w tym samym pliku Pythona (rozszerzenie `.py`). Opis powinien być na początku pliku w formie komentarza (w pierwszej linii w komentarzu powinno być imię i nazwisko studenta). Opis nie musi być długi—wystarczy kilka zdań, jasno opisujących ideę algorytmu. Implementacja musi być zgodna z szablonem kodu źródłowego dostarczonym wraz z zadaniem. Niedopuszczalne jest w szczególności:

1. zmienianie nazwy funkcji implementującej algorytm, listy jej argumentów, lub nazwy pliku z rozwiązaniem,
2. modyfikowanie testów dostarczonych wraz z szablonem,
3. wypisywanie na ekranie jakichkolwiek napisów innych niż wypisywane przez dostarczony kod (ew. napisy dodane na potrzeby diagnozowania błędów należy usunąć przed wysłaniem zadania).

Dopuszczalne jest natomiast:

1. korzystanie z następujących elementarnych struktur danych: krotka, lista, słownik, kolejka `collections.deque`, kolejka priorytetowa (`queue.PriorityQueue`, `heapq`),
2. korzystanie ze struktur danych dostarczonych razem z zadaniem (jeśli takie są).
3. korzystanie z wbudowanych funkcji sortujących (należy założyć, że mają złożoność  $O(n \log n)$ ).

Wszystkie inne algorytmy lub struktury danych wymagają implementacji przez studenta. Dopuszczalne jest oczywiście implementowanie dodatkowych funkcji pomocniczych w pliku z szablonem rozwiązania.

Zadania niezgodne z powyższymi ograniczeniami otrzymają ocenę 0 punktów. Rozwiązania w innych formatach (np. `.PDF`, `.DOC`, `.PNG`, `.JPG`) z definicji nie będą sprawdzane i otrzymają ocenę 0 punktów, nawet jeśli będą poprawne.

### Testowanie rozwiązań

Żeby przetestować rozwiązanie zadania należy wykonać polecenie: `python3 egz3b.py`

<b>Szablon rozwiązania:</b>	egz3b.py
<b>Złożoność akceptowalna (2pkt):</b>	$O(n^2)$
<b>Złożoność wzorcowa (+2pkt):</b>	$O(n \log n)$

Dany jest zbiór  $P = \{[a_0, b_0], \dots, [a_{n-1}, b_{n-1}]\}$  zawierający  $n$  parami różnych przedziałów domkniętych. Mówimy, że para przedziałów jest fajna jeśli albo jeden z nich zawiera się w drugim, albo są rozłączne. Jeśli para przedziałów nie jest fajna, to mówimy na nią niefajna.

**Przykład.** Rozważmy trzy przypadki:

1. Para przedziałów  $[1, 3]$  i  $[6, 7]$  jest fajna, bo przedziały są rozłączne.
2. Para przedziałów  $[2, 6]$  i  $[4, 6]$  jest fajna, bo drugi zawiera się w pierwszym.
3. Para przedziałów  $[1, 8]$  i  $[5, 10]$  jest niefajna, bo ani nie są rozłączne, ani jeden nie zawiera się w drugim.

Zadanie polega na implementacji funkcji:

```
uncool( P )
```

która na wejściu otrzymuje zbiór przedziałów  $P$  (w postaci listy, gdzie każdy element jest postaci  $[a_i, b_i]$  i opisuje przedział domknięty) i zwraca parę indeksów  $(i, j)$  takich, że para przedziałów  $P[i], P[j]$  jest niefajna (można założyć, że taka para przedziałów zawsze istnieje).

**Przykład.** Dla wejścia:

```
#      0      1      2      3      4      5
P =[ [1,3], [6,7], [2,6], [4,6], [1,8], [5,10] ]
```

prawidłowym wynikami są, między innymi,  $(2, 1)$  oraz  $(4, 5)$ .