Algorytmy i Struktury Danych
Zadanie 3 (18.III.2024)

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. korzystanie z wbudowanych funkcji sortujących,
2. korzystanie z zaawansowanych struktur danych (np. słowników, zbiorów),
3. zmienianie nazwy funkcji implementującej algorytm, listy jej argumentów, lub nazwy pliku z rozwiązaniem,
4. modyfikowanie testów dostarczonych wraz z szablonem,
5. 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.

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: `python zad3.py`

Zadanie offline 3.

Szablon rozwiązania: `zad3.py`

Dany jest zbiór $P = \{p_1, \ldots, p_n\}$ punktów na płaszczyźnie. Współrzędne punktów to liczby naturalne ze zbioru $\{1, \ldots, n\}$. Mówimy, że punkt $p_i = (x_i, y_i)$ dominuje punkt $p_j = (x_j, y_j)$ jeśli zachodzi:

$$x_i > x_j \text{ oraz } y_i > y_j.$$

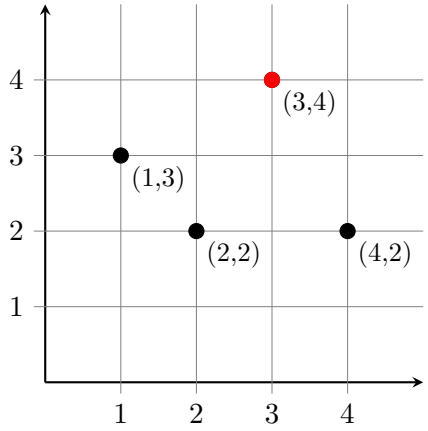
Siłą danego punktu jest to ile punktów dominuje. Zadanie polega na implementacji funkcji:

```
dominance( P )
```

która na wejściu otrzymuje listę `P` zawierającą n punktów (każdy reprezentowany jako para liczb ze zbioru $\{1, \ldots, n\}$) i zwraca siłę najsilniejszego z nich. Funkcja powinna być możliwie jak najszybsza.

Przykład. Dla wejścia:

```
P = [(1,3),
     (3,4),
     (4,2),
     (2,2)]
```



wynikiem jest 2. Punkt o współrzędnych $(3, 4)$ dominuje punkty o współrzędnych $(1, 3)$ oraz $(2, 2)$.