

Algorytmy i Struktury Danych
Egzamin/Zaliczenie 1 (13.VII.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 (można 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 egz1b.py`

| | |
|-----------------------------------------------------------------|------------------------------------------------------------|
| Szablon rozwiązania: | egz1b.py |
| Złożoność akceptowalna (3.0pkt): | złożoność wielomianowa względem n i E (np. $O(nE^2)$) |
| Złożoność wzorcowa (+1.0pkt): | $O(nE)$ |
| Gdzie n to liczba planet a E to pojemność zbiornika paliwa. | |

Przedstawiciel Gwiezdnej Floty podróżuje z planety A do planety B. Po drodze będzie musiał wylądować na innych planetach w celu uzupełnienia paliwa. Cena paliwa na każdej planecie może być inna. Dodatkowo na każdej planecie jest teleport, który może przenieść go na jedną z kolejnych planet. Niestety, w tym celu jego statek kosmiczny musi mieć pusty zbiornik paliwa. Zaproponuj i zaimplementuj algorytm, który oblicza minimalny koszt pokonania trasy z planety A do B.

Na trasie przedstawiciela Gwiezdnej Floty znajduje się n planet opisanych w tablicach D, C i T:

- D[i] to odległość i -tej planety od planety A wyrażona w latach świetlnych wzdłuż trasy przelotu. Planety uporządkowane są w kolejności rosnącej odległości od A. Nie dopuszczamy sytuacji, w której dwie planety mają tę samą odległość od A.
- C[i] to cena jednej tony paliwa na planecie numer i ,
- T[i] to para postaci (j, p) , gdzie j to numer planety, na którą można dostać się teleportem z planety i (zawsze zachodzi $j \geq i$, gdzie $j = i$ oznacza, że teleport na tej planecie nie działa) a p to cena skorzystania z teleportu.

Planeta A ma numer 0 a planeta B ma numer $n - 1$. Statek kosmiczny potrzebuje tony paliwa na pokonanie każdego roku świetlnego. Pojemność zbiornika statku kosmicznego wynosi E ton. Nie ma obowiązku tankowania paliwa do pełna. Zakładamy, że E oraz wszystkie elementy tablic D, C i T to liczby naturalne. Można założyć, że rozwiązanie istnieje, t.j. da się przelecieć z A do B zgodnie z warunkami zadania.

Algorytm należy zaimplementować jako funkcję:

```
planets( D, C, T, E )
```

Proszę uzasadnić poprawność zaproponowanego algorytmu i oszacować jego złożoność obliczeniową.

Przykład. Dla wejścia:

```
#          0          1          2          3
D = [      0,        5,       10,       20]
C = [      2,        1,        3,        9]
T = [(2,3), (3,7), (2,10), (3,10)]
E = 10
```

wynikiem jest 17, co odpowiada zatankowaniu 5 ton paliwa na planecie A (czyli planecie 0, koszt 10), przelotowi na planetę 1 oraz skorzystaniu z teleportu z planety 1 na planetę 3 (czyli planetę B, koszt 7).

Podpowiedź. Proszę rozważyć funkcję $f(i, b)$, która daje minimalny koszt znalezienia się na planecie i mając b ton paliwa. Obliczanie tej funkcji w odpowiednio przemyślany sposób pozwala uzyskać złożoność wzorcową.