

Szablon rozwiązania	egzP9a.py
Złożoność akceptowalna (1.5pkt):	$O(q\sqrt{n})$
Złożoność wzorcowa (+2.5pkt):	$O(q\log n)$ , gdzie $n$ to rozmiar tablicy, a $q$ to ilość zapytań

W wakacje na wydział WIEiT przyszedł magiczny superkomputer ASD-0x1000101, który jednak ma pewne ograniczenia. Pozwala on na korzystanie z dwóch operacji i tablicy TAB zawierającą nieujemne liczby całkowite, początkowo wypełnioną zerami. Funkcja `set(index, val)` w języku Python wykonałaby operację analogiczną do `TAB[index] = val`. Funkcja `get(index)` w języku Python wykonałaby operację analogiczną do `return TAB[index]`. Bajtek bardzo zainteresował się nowym nabytkiem, jednak kiedy zapytał dziekana czy wolno mu spróbować coś napisać na ASD-0x1000101, niestety usłyszał on *"Proszę Pana, taka przyjemność to będzie Pana kosztować 50zł, takie rzeczy dopiero po WDI"*. Rozczarowany Bajtek poszedł szukać rozwiązań do koła naukowego - BIT-u. Koło naukowe jest organizacją non-profit, zatem nie będzie musiał on nic płacić, natomiast w ramach blokowania zasobów które mogłyby być wykorzystywane do kopania kryptowalut, za każdą operację na ASD-0x1000101 Bajtek będzie musiał zapłacić jednego BIT COIN-a, które może zdobyć za wykonywanie działalności na rzecz koła. Bajtek stwierdził, że algorytm na którym chce przetestować działanie ASD-0x1000101 musi być tak szybki jak się da, ponieważ woli spędzić więcej czasu wychodząc na Miasteczku Studenckim i grając we flanki.

Zadanie polega na zaimplementowaniu funkcji:

```
def ASD( T, p, Q, n )
```

która zwraca sumę wszystkich zapytań typu **B**, wykonując przy tym operacje jak najniższym kosztem (tj. jak najmniejszą ich liczbę), przy następujących założeniach:

1. Zmienna **T** nie jest tablicą, lecz symulatorem superkomputera. Daje on dostęp do funkcji `T.set(index, val)` która zmienia wartość `TAB[index]` na wartość zmiennej `val`, oraz funkcji `T.get(index)` która zwraca wartość `TAB[index]`

2. Każde z zapytań w tablicy **Q** jest albo typu **A** tj. w postaci krotki `(0, index, val)` – czyli powinno powodować zwiększenie `TAB[index]` o wartość `val`, albo typu **B**, tj. w postaci krotki `(1, a, b)` – czyli powinno zwracać sumę liczb w tablicy na przedziale od `a` do `b`.

3. Zmienna **p** wyraża rozmiar wewnętrznej tablicy superkomputera (długość tablicy TAB w nim zapisanej). Można założyć, że cała tablica, do której Bajtek wysyła zapytania, o rozmiarze **n**, zmieści się na superkomputerze. W większości testów rozmiar wewnętrznej tablicy superkomputera jest dwukrotnie większy niż liczba elementów tablicy Bajtka. W każdym teście rozmiar wewnętrznej tablicy superkomputera jest większy, niż liczba elementów tablicy Bajtka.

**Uwaga.** Zadeklarowanie własnej tablicy w celu rozwiązania zadania, z pominięciem zastosowania tablicy znajdującej się w superkomputerze **T** spowoduje wyzerowanie punktów przez testy.

**Podpowiedź.** Format zadania jest dość nietypowy (także ze względu na sposób działania testów), w takiej formie żadne zadanie raczej nie powinno się pojawić na egzaminie. Zadanie ma na celu sprawdzenie zrozumienia struktury **Drzewa Przedziałowego**

Szablon rozwiązania

egzP9b.py

Złożoność akceptowalna (1.5pkt):

$O(n^2)$

Złożoność wzorcowa (+2.5pkt):

$O(n \log n)$ , gdzie  $n$  to łączna liczba krawędzi grafu ( $V < E$ )

W Polsce od początku XXI wieku rozbudowywana była sieć dróg krajowych i autostrad. Większość miast (w zadaniu wyrażona jako wierzchołki grafu) jest połączona ze sobą różnymi drogami, zazwyczaj więcej niż jedna droga łączy dwa miasta. Ze względu na organizację ruchu wszystkie zaznaczone na mapie drogi są jednokierunkowe. Dodatkowo część dróg aktualnie jest remontowana i nie można nią przejechać (oczywiście remontowane drogi także były jednokierunkowe). Dyrektor zarządu dróg postanowił, że osobiście sprawdzi wszystkie drogi, jednak aby zoptymalizować swoją trasę, każdą z nich przejedzie dokładnie jeden raz. Zaczyna on swój objazd w mieście 0 i w tym samym mieście zamierza go zakończyć.

Zadanie polega na zaimplementowaniu funkcji:

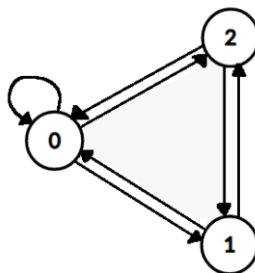
```
def dyrektor( G, R )
```

która zwróci kolejność, w jakiej dyrektor zarządu dróg odwiedzi Polskie miasta zakładając, że

- Graf **G** jest multigrafem skierowanym reprezentującym sieć drogową wyrażonym w postaci listy sąsiedztwa (należy zauważyć, że dla wierzchołki w obrębie list mogą się więc powtarzać)
- Reprezentacja grafu **R** jest analogiczna do grafu **G**, jednak zawiera on jedynie remontowane drogi (Za to graf **G** zawiera wszystkie drogi, niezależnie od tego, czy są remontowane)

Rozważmy następujące dane:

```
G = [  
  [1, 0, 2],  
  [2, 0],  
  [1, 0]  
]  
R = [  
  [0],  
  [],  
  []  
]
```



Wywołanie `dyrektor( G, R )` powinno zwrócić wynik np. **[0, 1, 2, 0, 2, 1, 0]**. Proszę zauważyć, że może być wiele poprawnych wyników. Testy zostały dobrane tak, aby zadanie zawsze dało się rozwiązać.