

# W DROGĘ!

## Wstęp

W Universum zapanował względny pokój. Żukoskoczki dzielnie odparły nagły najazd krwiożerczych ważkolotów, a fabryka amunicji *BitBullet* do dziś przyjmuje listy gratulacyjne od najwyższych rangą urzędników wojskowych. Oprócz świętowania, nadszedł czas dogłębnych analiz i wyciągania wniosków. Czy wszystko poszło tak jak trzeba?

„Rozwóz amunicji: STRATY” – raport końcowy okazał się bezlitosny. Pracownicy fabryki doskonale poradzili sobie z upakowaniem amunicji do kontenerów. Niestety, plan dostarczenia amunicji szwankował i jego wdrożenie przyniosło straty...

## Zadanie

Waszym zadaniem jest zaplanowanie rozwoju amunicji na wypadek wznowienia działań wojennych. Należy obsłużyć  $C$  klientów rozmieszczonych w przestrzeni dwuwymiarowej – pozycja  $i$ -tego klienta jest dana jako punkt  $(x_i, y_i)$ . Fabryka *BitBullet*, znajdująca się w punkcie  $(m_x, m_y)$ , jest punktem startowym i końcowym każdego z żukarów – samochodów używanych do dystrybucji amunicji. Żukoskoczki przezornie budują swoje bazy z dala od innych, dlatego lokalizacje *BitBullet* i klientów fabryki są zawsze unikalne. Wszystkie żukary mają tę samą pojemność, równą  $Q$  jednostek amunicji i wyruszają z fabryki w momencie  $t_0 = 0$ . Klienci *BitBullet* są świetnie zorganizowani i tego samego wymagają od swojego dostawcy.

Każdy  $i$ -ty klient czeka na  $d_i$  jednostek amunicji w swoim slotcie czasowym  $[b_i, e_i]$ . Żukar, który przyjeżdża do klienta przed otwarciem jego slotu musi czekać do czasu  $b_i$  – dopiero wtedy może wjechać na platformę rozładunkową. Instalacja żukara na platformie jest wyjątkowo trudna, dlatego każdy klient może zostać odwiedzony tylko raz. Jeśli towar jest dostarczony w czasie trwania slotu, to rozładunek rozpoczyna się natychmiast. Rozładunek amunicji trwa  $s_i$  i może zakończyć się poza oknem czasowym. Czas przejazdu między dowolną parą punktów (dwoma klientami lub klientem i fabryką) jest równy odległości między tymi punktami w metryce miejskiej (suma modułów różnic współrzędnych).

Nierespektowanie wymagań któregośkolwiek z klientów (przyjazd później niż w  $e_i$ ) wiąże się z ogromnymi karami finansowymi i utratą zaufania pozostałych kontrahentów. *BitBullet* nie może sobie na to pozwolić. Dodatkowo koszty eksploatacji żukarów są bardzo wysokie, a żuliwo (paliwo służące do napędzania żukarów) wciąż drożeje.

## Dane wejściowe

Zestawy testowe znajdują się w plikach `roads*.in`.

Pierwsza linia zestawu testowego zawiera dwie (oddzielone pojedynczymi spacjami) liczby naturalne:  $C$  – oznaczającą liczbę klientów, oraz  $Q$  – oznaczającą maksymalną pojemność każdego z pojazdów. Druga linia zawiera dwie (oddzielone pojedynczymi spacjami) liczby naturalne  $m_x$  oraz  $m_y$ , oznaczające pozycję fabryki *BitBullet* w układzie współrzędnych kartezjańskich. Każda  $i$ -ta z kolejnych  $C$  linii składa się z siedmiu liczb naturalnych:  $ID_i, x_i, y_i, b_i, e_i, d_i, s_i$ , oznaczających kolejno identyfikator klienta, jego pozycję, okno czasowe, liczbę jednostek towaru, na którą czeka, oraz czas rozładunku.

$$\begin{aligned} 0 < C, Q, ID &\leq 10^4 \\ 0 \leq m_x, m_y, x_i, y_i &\leq 5 \cdot 10^4 \\ 0 \leq b_i, e_i, s_i &\leq 10^5 \\ 0 < d_i &\leq Q \end{aligned}$$

## Dane wyjściowe

Dane wyjściowe powinny zawierać w pierwszej linii liczbę  $K$ , oznaczającą liczbę tras (równoważnie – liczbę potrzebnych żukarów), oraz liczbę  $T$ , będącą sumaryczną długością wszystkich tras nadesłanego rozwiązania. Każda z kolejnych  $K$  linii opisuje jedną trasę. Trasa powinna być podana jako ciąg identyfikatorów kolejno odwiedzonych klientów.

## Przykład

Dla danych wejściowych:

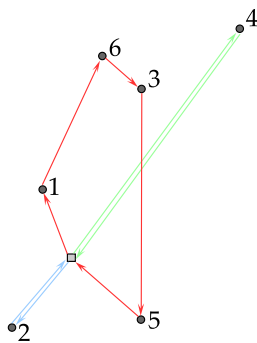
```
6 20
9 9
1 7 13 0 10 7 0
2 5 5 3 9 2 2
3 14 17 1 25 4 1
4 19 22 3 24 1 3
5 15 6 40 45 2 5
6 11 19 1 16 5 2
```

Jedno z możliwych rozwiązań to:

```
3 104
2
1 6 3 5
4
```

## Objaśnienie przykładu

Wszyscy klienci zostali obsłużeni przez 3 żukary:



- Trasa 1: *BitBullet* → 2 → *BitBullet*
  - $t = t_0 = 0$ : wyruszenie z *BitBullet* do klienta 2
  - $t = 8$ : dotarcie do klienta 2, rozładunek
  - $t = 10$ : wyruszenie do *BitBullet*
  - $t = 18$ : dotarcie do *BitBullet*
- Trasa 2: *BitBullet* → 1 → 6 → 3 → 5 → *BitBullet*
  - $t = t_0 = 0$ : wyruszenie z *BitBullet* do klienta 1
  - $t = 6$ : dotarcie do klienta 1, rozładunek, wyruszenie do klienta 6
  - $t = 16$ : dotarcie do klienta 6, rozładunek

- $t = 18$ : wyruszenie do klienta 3
- $t = 23$ : dotarcie do klienta 3, rozładunek
- $t = 24$ : wyruszenie do klienta 5
- $t = 36$ : dotarcie do klienta 5
- $t = 40$ : rozładunek
- $t = 45$ : wyruszenie do *BitBullet*
- $t = 54$ : dotarcie do *BitBullet*

- Trasa 3: *BitBullet* → 4 → *BitBullet*

- $t = t_0 = 0$ : wyruszenie z *BitBullet* do klienta 4
- $t = 23$ : dotarcie do klienta 4, rozładunek
- $t = 26$ : wyruszenie do *BitBullet*
- $t = 49$ : dotarcie do *BitBullet*

$$\begin{aligned}
 T = & |9 - 5| + |9 - 5| + |5 - 9| + |5 - 9| + \\
 & + |9 - 7| + |9 - 13| + |7 - 11| + |13 - 19| + |11 - 14| + \\
 & + |19 - 17| + |14 - 15| + |17 - 6| + |15 - 9| + |6 - 9| + \\
 & + |9 - 19| + |9 - 19| + |19 - 9| + |19 - 9| = 104
 \end{aligned}$$

$$S = \frac{6}{3} + \frac{142}{104} = 3.365$$

## Ocena

Jeśli spełnione są wszystkie poniższe warunki:

- dane wyjściowe są poprawnie sformatowane,
- liczba wykorzystanych żukarów ( $K$ ) jest mniejsza lub równa liczbie klientów ( $C$ ),
- pojemność żadnego z żukarów nie została przekroczona,
- rozładowywanie amunicji u każdego klienta rozpoczęło się w trakcie trwania jego okna czasowego,
- każdy klient został odwiedzony tylko raz,
- sumaryczna długość tras ( $T$ ) została poprawnie wyliczona,

to ocena za dany zestaw jest równa wartości  $S = \frac{C}{K} + \frac{T_0}{T}$  (zaokrąglonej do trzech miejsc po przecinku), gdzie wartość  $T_0$  jest równa sumarycznej długości wszystkich tras dla rozwiązania, w którym  $K = C$ . W przeciwnym wypadku ocena wynosi 0.