Szablon rozwiązania	`egzP5a.py`
Złożoność akceptowalna (1.5pkt):	$O(n^2)$
Złożoność wzorcowa (+2.5pkt):	$O(n)$, gdzie ***n*** to liczba działek

W jednym z kanadyjskich dystryktów, farmerzy uprawiali kukurydzę. Poważny amerykański inwestor pracujący w jednej z największych firm, postanowił wykupić część tych działek, aby zamiast kukurydzy, zbudować tam szklarnie do uprawy cytrusów. Zdając sobie sprawę, że wśród farmerów znajdują się zazdrośnicy, postanowił, że od każdego farmera u którego zainwestuje, kupi fragment działki o dokładnie takim samym polu powierzchni. Ponadto, aby zminimalizować koszty przeznaczone na notariusza, zależy mu na tym, aby wszystkie działki, których fragmenty wykupi, znajdywały się kolejno po sobie w księgach wieczystych. Jako doradca finansowy inwestora, zostałeś poproszony o obliczenie, jakie jest maksymalne możliwe łączne pole powierzchni części działek, w które zainwestuje, przy powyższych założeniach.

Zadanie polega na zaimplementowaniu funkcji:

```
inwestor( T )
```

która obliczy to największe możliwe łączne pole, przy założeniu, że tablica **T** zawiera pola powierzchni działek wyrażone w hektarach oraz, że kolejność elementów w tablicy jest zgodna z kolejnością występowania tych działek w księgach wieczystych. Dla ułatwienia przyjmujemy, że nie ma działek o zerowym polu powierzchni.

Rozważmy następujące dane:

```
T = [2, 1, 5, 6, 2, 3]
```

Wywołanie funkcji `inwestor( T )` powinno zwrócić wynik **10** (Inwestujemy w działki o polach powierzchni **5 ha** oraz **6 ha**, wykupując z obydwu po **5 ha**)

Podpowiedź. Czy informacja o tym, jakie jest najmniejsze pole spośród pewnego zbioru działek, posiadana dla każdego dowolnego przedziału (a, b) ułatwiłaby rozwiązanie tego zadania?

Szablon rozwiązania	`egzP5b.py`
Złożoność akceptowalna (1.5pkt):	O(n^2)
Złożoność wzorcowa (+2.5pkt):	O($n\log n$), gdzie ***n*** to liczba sprzedanych biletów

Sieć kolejowa w Polsce w ostatnich latach rozwija się bardzo dynamicznie. W nowej ofercie biletowej, która została dopuszczona do sprzedaży miesiąc temu, przedstawiona została nowa mapa dostępnych połączeń. Jako, iż okazały się one atrakcyjne, na każde z nich został zakupiony przynajmniej jeden bilet. PKP Intercity prowadzi bardzo dokładne statystyki, zapisując informacje o każdym sprzedanym bilecie. Jako, że spółka posiada wolne środki, planuje potężny remont jednego z dworców kolejowych, co na pewien okres wykluczy go zarówno z obsługi podróżnych, jak i pociągów (w skrócie – żaden pociąg nie będzie mógł przez niego przejechać). Samo tymczasowe wyłączenie dworca nie jest problemem, ponieważ uruchomiona zostanie zastępcza komunikacja autobusowa, aczkolwiek nie może dojść do sytuacji, w której spowoduje to brak możliwości przejazdu między innymi dworcami. W związku z tym firma musiała sporządzić listę potencjalnych dworców, których remont nie spowoduje tego typu problemu.

Zadanie polega na zaimplementowaniu funkcji:

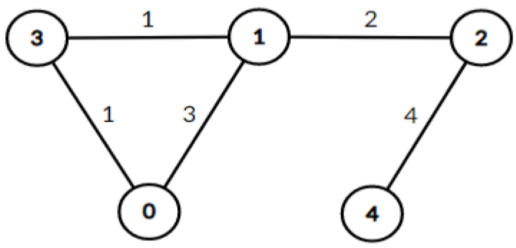
```
koleje( B )
```

która obliczy liczbę dworców na wspomnianej liście, przy następujących założeniach:

- Tablica **B** zawiera listę wszystkich sprzedanych biletów od czasu aktualizacji oferty. Każdy bilet jest w postaci krotki *(p, k)* gdzie *p* to indeks stacji początkowej, a *k* to indeks stacji końcowej.
- Dla celów oszacowania złożoności obliczeniowej należy przyjąć, że największy indeks stacji jest mniejszy od łącznej ilości sprzedanych biletów.

Rozważmy następujące dane:

```
B = [
  (3, 1), (0, 1), (4, 2),
  (1, 2), (0, 1), (2, 4),
  (2, 4), (0, 3), (2, 4),
  (1, 0), (2, 1)
]
```



Wywołanie `koleje( B )` powinno zwrócić wynik **2**. Po zrzutowaniu każdego zakupionego biletu jako połączenia (co widać na załączonym obrazku) można zauważyć, że zarówno usunięcie *dworca **1***, jak i *dworca **2*** spowodowałoby problemy z przejazdem. Pozostałe dworce można w bezpieczny sposób wyremontować.