Park fontann (Fountain Park)

W pobliskim parku zbudowano n **fontann** etykietowanych od 0 do $n-1$. Reprezentujemy je jako punkty na dwuwymiarowej powierzchni. Fontanna i ($0 \leq i \leq n-1$) jest punktem o współrzędnych $(x[i], y[i])$ gdzie $x[i]$ oraz $y[i]$ są **całkowitymi liczbami parzystymi**. Każda fontanna jest w innym punkcie.

Architekt Tymoteusz planuje połączyć fontanny **ścieżkami**, przy których będą ustawiane **ławki** - dokładnie jedna przy każdej ścieżce. Ścieżka to **pionowa** lub **pozioma** krawędź długości 2 łącząca dwie fontanny. Ścieżki mają być tak zaprojektowane, aby można było po nich przejść od każdej fontanny do każdej innej. Na razie w parku nie ma żadnych ścieżek.

Z każdą ścieżką musi być skojarzona **dokładnie jedna** ławka. Możemy sobie wyobrazić, że siedząc na ławce patrzymy na tę jedyną ścieżkę. Każda ławka musi mieć współrzędne (a, b) takie, że a oraz b są **nieparzystymi liczbami całkowitymi**. W jednym punkcie może się znajdować **co najwyżej jedna** ławka. Ławka o współrzędnych (a, b) może być skojarzona ze ścieżką, jeśli **obie** współrzędne tej ścieżki są wśród $(a-1, b-1)$, $(a-1, b+1)$, $(a+1, b-1)$ lub $(a+1, b+1)$. Na przykład ławka umieszczona w $(3, 3)$ może być skojarzona tylko z jedną z czterech ścieżek: $(2, 2)$ – $(2, 4)$, $(2, 4)$ – $(4, 4)$, $(4, 4)$ – $(4, 2)$, $(4, 2)$ – $(2, 2)$.

Pomóż Tymoteuszowi rozstrzygnąć, czy jest w ogóle możliwe zaprojektowanie ścieżek i skojarzenie z nimi ławek tak, aby powyższe warunki były spełnione. Jeśli tak, to przedstaw propozycję takiego rozwiązania. Gdyby było możliwe więcej niż jedno rozwiązanie, zaproponuj dowolne z nich.

Szczegóły implementacyjne

Powinieneś zaimplementować następującą funkcję:

```
int construct_roads(int[] x, int[] y)
```

- x, y: dwie tablice długości n. Dla każdego i ($0 \leq i \leq n-1$), fontanna i ma współrzędne $(x[i], y[i])$, gdzie $x[i]$ oraz $y[i]$ są całkowitymi liczbami parzystymi.
- Jeśli zaprojektowanie ścieżek i ławek jest możliwe, ta funkcja powinna dokładnie raz wywołać procedurę `build` (patrz poniżej) aby podać jakieś rozwiązanie, po czym powinna przekazać wynik 1.
- W przeciwnym razie procedura powinna przekazać wynik 0 bez wywołania procedury `build`.
- Ta funkcja będzie wywołana dokładnie raz.

Opisana funkcja może wywołać następującą procedurę zgłaszającą poprawne rozmieszczenie ścieżek i ławek.

```
void build(int[] u, int[] v, int[] a, int[] b)
```

- Niech m będzie liczbą ścieżek użytych w tej konstrukcji.
- u, v: dwie tablice długości m określające projektowane ścieżki. Te ścieżki mają etykiety od 0 do $m-1$. Dla każdego j ($0 \leq j \leq m-1$), ścieżka j łączy fontanny $u[j]$ oraz $v[j]$. Każda ścieżka musi być pionowym bądź poziomym odcinkiem długości 2. Każde dwie różne ścieżki mogą mieć co najwyżej jeden punkt wspólny (fontannę). Gdy zostaną zbudowane, powinno dać się dojść idąc po nich z każdej fontanny do każdej innej.
- a, b: dwie tablice długości m określające położenia ławek. Dla każdego j ($0 \leq j \leq m-1$), j-ta ławka ma współrzędne $(a[j], b[j])$ i jest skojarzona ze ścieżką j. Żadne dwie ławki nie mają tych samych współrzędnych.

Przykłady

Przykład 1

Rozważmy następujące wywołanie:

```
construct_roads([4, 4, 6, 4, 2], [4, 6, 4, 2, 4])
```

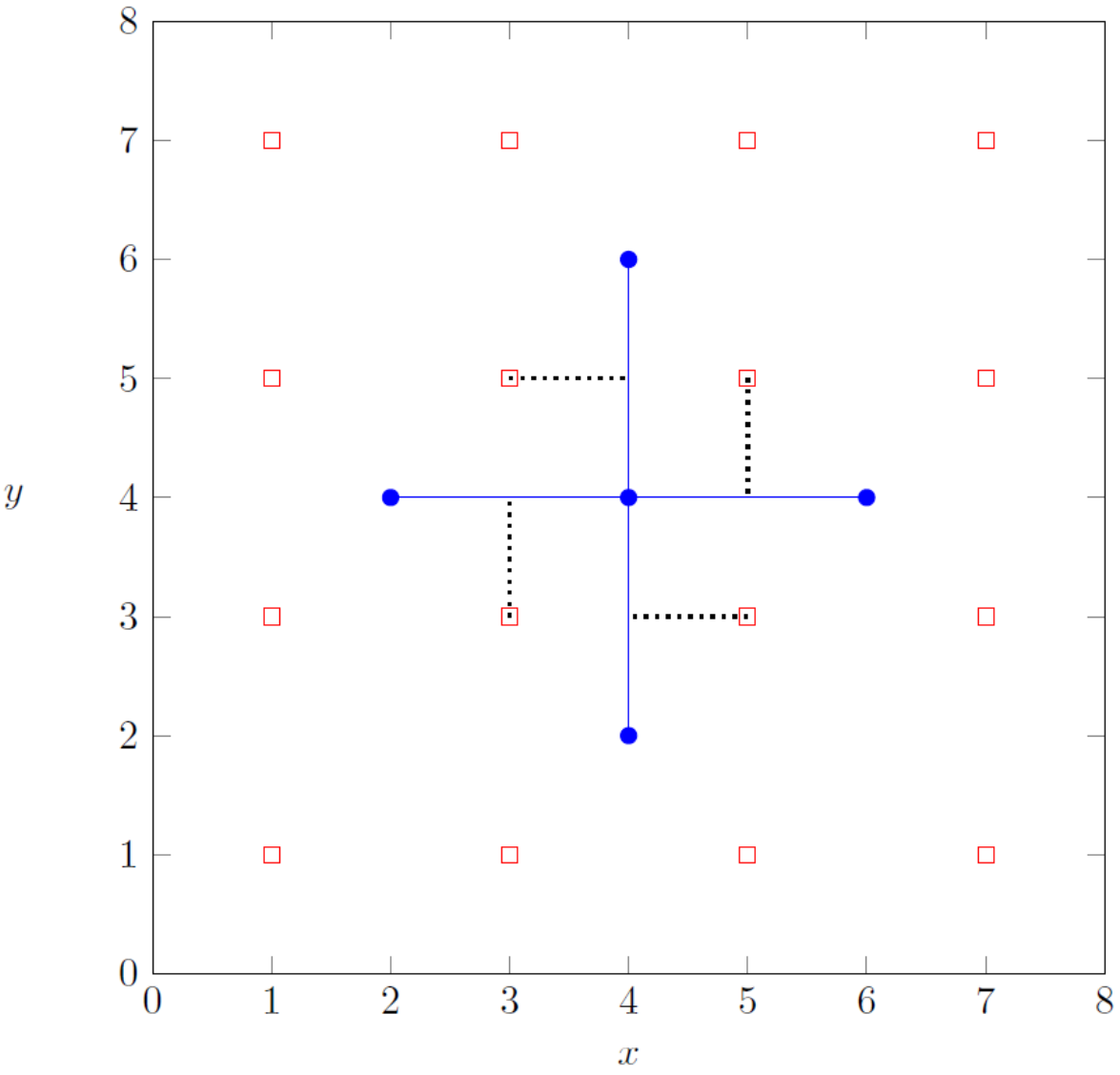
Oznacza ono, że mamy 5 fontann:

- fontanna 0 ma współrzędne $(4, 4)$,
- fontanna 1 ma współrzędne $(4, 6)$,
- fontanna 2 ma współrzędne $(6, 4)$,
- fontanna 3 ma współrzędne $(4, 2)$,
- fontanna 4 ma współrzędne $(2, 4)$.

Można zaprojektować następujące 4 ścieżki i skojarzone z nimi ławki:

Etykieta ścieżki	Etykiety fontann, które ścieżka łączy	Współrzędne skojarzonej ławki
0	$0, 2$	$(5, 5)$
1	$0, 1$	$(3, 5)$
2	$3, 0$	$(5, 3)$
3	$4, 0$	$(3, 3)$

To rozwiązanie odpowiada następującemu rysunkowi:



Aby podać rozwiązanie, funkcja `construct_roads` powinna wywołać

- `build([0, 0, 3, 4], [2, 1, 0, 0], [5, 3, 5, 3], [5, 5, 3, 3])` i zakończyć działanie przekazując wartość 1.

Zwróć uwagę na to, że w tym przypadku jest wiele rozwiązań spełniających warunki zadania. Wszystkie uznajemy za poprawne. Na przykład również poprawne byłoby wywołanie `build([1, 2, 3, 4], [0, 0, 0, 0], [5, 5, 3, 3], [5, 3, 3, 5])` i przekazanie wartości 1.

Przykład 2

Rozważmy następujące wywołanie:

```
construct_roads([2, 4], [2, 6])
```

Fontanna 0 ma współrzędne $(2, 2)$, a fontanna 1 ma współrzędne $(4, 6)$. Ponieważ nie da się zaprojektować ścieżek spełniających wymagania, funkcja `construct_roads` powinna przekazać wynik 0 nie wywołując ani razu procedury `build`.

Ograniczenia

- $1 \leq n \leq 200\,000$
- $2 \leq x[i], y[i] \leq 200\,000$ (dla każdego $0 \leq i \leq n-1$)
- $x[i]$ oraz $y[i]$ są całkowite i parzyste (dla każdego $0 \leq i \leq n-1$).
- Położenie każdych dwóch fontann jest różne.

Podzadania

1. (5 punktów) $x[i] = 2$ (dla każdego $0 \leq i \leq n-1$)
2. (10 punktów) $2 \leq x[i] \leq 4$ (dla każdego $0 \leq i \leq n-1$)
3. (15 punktów) $2 \leq x[i] \leq 6$ (dla każdego $0 \leq i \leq n-1$)
4. (20 punktów) Istnieje co najwyżej jedna możliwość zaprojektowania ścieżek tak, żeby dało się po nich przejść między każdymi dwiema fontannami.
5. (20 punktów) Nie istnieją fontanny, których położenia tworzą kwadrat 2 na 2.
6. (30 punktów) Brak dodatkowych ograniczeń.

Przykładowa sprawdzaczka

Przykładowa sprawdzaczka czyta wejście w następującym formacie:

- wiersz 1: n
- wiersz $2+i$ ($0 \leq i \leq n-1$): $x[i]$ $y[i]$

Wyjście przykładowej sprawdzaczki jest następujące:

- wiersz 1: wynik wywołania `construct_roads`

Jeśli wynik wywołania `construct_roads` to 1 i wywołane zostało `build(u, v, a, b)`, to sprawdzaczka dodatkowo wypisuje

- wiersz 2: m
- wiersz $3+j$ ($0 \leq j \leq m-1$): $u[j]$ $v[j]$ $a[j]$ $b[j]$