

Kelner w naleśnikarni przynosi gościom do stolików talerze z naleśnikami.

Naleśniki ułożone są byle jak, więc czasami spadają z tacy (i trzeba je podnosić z podłogi tak, żeby klienci nie widzieli). Kelner wpadł więc na pomysł, że będzie je unieruchamiał za pomocą wykałaczki (jednej, bo tyle ma).

Naleśniki są kołami o różnych promieniach. Zadanie sprowadza się więc do znalezienia punktu wspólnego wszystkich kół lub stwierdzenia, że takiego nie ma. Zwracamy dowolny punkt wspólny kół (może leżeć na brzegu) lub null, jeśli punkt wspólny nie istnieje.

Punktacja:

- metoda `GetIntersectionType` 1 pkt.
- metoda `FindCommonPoint` wraz z ewentualnymi metodami pomocniczymi
2 pkt. za wersję działającą w czasie n^3
3 pkt. za wersję działającą w czasie n^2

Podpowiedzi:

Wersja za 2 pkt (złożoność n^3)

- 1) Przeprowadzić przetwarzanie wstępne eliminujące koła całkowicie zawierające w sobie inne koła (a z duplikatów pozostawiając jeden egzemplarz), oczywiście po wykryciu kół rozłącznych można od razu zakończyć algorytm.
- 2) Badając wszystkie pary okręgów wyznaczyć wszystkie punkty przecięcia (lub styczności) dwóch okręgów. Jeśli zbiór punktów wspólnych wszystkich kół jest niepusty to zawiera co najmniej jeden z tych punktów przecięcia.

Wersja za 3 pkt (złożoność n^2)

- 1) Przeprowadzić przetwarzanie wstępne eliminujące koła całkowicie zawierające w sobie inne koła (a z duplikatów pozostawiając jeden egzemplarz), oczywiście po wykryciu kół rozłącznych można od razu zakończyć algorytm.
- 2) Badając wszystkie pary kół wyznaczyć najbardziej prawy (o największej współrzędnej x) punkt części wspólnej każdej pary kół (zastanowić się jakie punkty mogą być kandydatami na taki najbardziej prawy punkt). Jeśli zbiór punktów wspólnych wszystkich kół jest niepusty to zawiera najbardziej lewy z tych najbardziej prawych punktów.

Należy pamiętać, że w arytmetyce zmiennopozycyjnej obliczenia nie są dokładne, więc mogą być potrzebne sprawdzenia z dokładnością do stałej `Program.epsilon`.