Jakub Radoszewski **Maciej Jaśkowski**
Treść zadania, Opracowanie Program

OI, Etap III, dzień pierwszy, 28.03.2007

Gazociągi

Koncern GazBit zamierza zdominować rynek gazowy w Bajtocji. Specjaliści umiejscowili już na mapie Bajtocji optymalne położenia punktów wydobycia gazu oraz stacji dystrybucji gazu — pozostało jeszcze tylko przyporządkować stacje do punktów wydobycia. Każda stacja dystrybucji ma być połączona z dokładnie jednym punktem wydobycia i odwrotnie, każdy punkt wydobycia z dokładnie jedną stacją.

GazBit specjalizuje się w budowie gazociągów prowadzących z punktów wydobycia do stacji dystrybucji w kierunku południowym i wschodnim — dokładniej, każdy wybudowany gazociąg ma (patrząc z lotu ptaka) kształt łamanej, której każda kolejna część biegnie w kierunku południowym lub wschodnim i jest prostopadła do poprzedniej. Zarząd koncernu zastanawia się, jak przyporządkować punktom wydobycia gazu stacje dystrybucji tak, by zminimalizować łączną długość koniecznych do wybudowania gazociągów. Przy planowaniu można pominąć problem przecinania się gazociągów — ich kolidujące fragmenty zostaną umieszczone na różnych głębokościach pod ziemią.

Zadanie

Napisz program, który:

- *wczyta ze standardowego wejścia planowane położenia punktów wydobycia i stacji dystrybucji gazu,*
- *wyznaczy takie przyporządkowanie stacji dystrybucji do punktów wydobycia, które pozwala na wybudowanie gazociągów o minimalnej łącznej długości,*
- *wypisze wynik na standardowe wyjście.*

Wejście

Pierwszy wiersz wejścia zawiera jedną liczbę całkowitą n *(*$2 \leqslant n \leqslant 50\,000$*), oznaczającą liczbę punktów wydobycia (równą liczbie stacji dystrybucji). Kolejne* n *wierszy zawiera po dwie liczby całkowite* x_i *oraz* y_i *(*$0 \leqslant x_i, y_i \leqslant 100\,000$ *dla* $1 \leqslant i \leqslant n$*), oddzielone pojedynczym odstępem i oznaczające współrzędne na mapie punktów wydobycia gazu. Przyjmujemy, że wraz z rosnącą współrzędną* x *poruszamy się na wschód, a wraz z rosnącą współrzędną* y *poruszamy się na północ. Następne* n *wierszy zawiera po dwie liczby całkowite* x'_j *oraz* y'_j *(*$0 \leqslant x'_j, y'_j \leqslant 100\,000$ *dla* $1 \leqslant j \leqslant n$*), oddzielone pojedynczym odstępem i oznaczające współrzędne na mapie stacji dystrybucji gazu. Zarówno punkty wydobycia, jak i stacje dystrybucji, numerujemy liczbami naturalnymi od 1 do* n *w kolejności występowania na wejściu. Żadna para współrzędnych nie powtórzy się w jednym zestawie danych wejściowych. Ponadto dla każdych danych wejściowych istnieje jakieś przyporządkowanie punktom wydobycia stacji dystrybucji, które można zrealizować za pomocą gazociągów idących tylko na południe i wschód.*

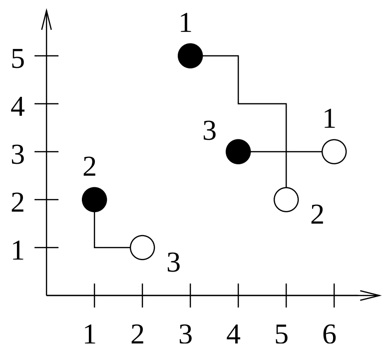
Wyjście

Pierwszy wiersz wyjścia powinien zawierać jedną liczbę całkowitą, oznaczającą minimalną sumaryczną długość wszystkich koniecznych do wybudowania gazociągów. Dalej na wyjściu powinien wystąpić przykładowy opis przyporządkowania stacji punktom wydobycia, który realizuje to minimum. Każdy z kolejnych n wierszy powinien zawierać dwie liczby całkowite, oddzielone pojedynczym odstępem i oznaczające numery punktu wydobycia i stacji dystrybucji, które powinny być połączone gazociągiem. Kolejność wypisywania przyporządkowań może być dowolna. Jeżeli istnieje wiele poprawnych rozwiązań, Twój program powinien wypisać jakiekolwiek z nich.

Przykład

Dla danych wejściowych:

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



poprawnym wynikiem jest:

```
9
2 3
1 2
3 1
```

Rozwiązanie

Kilka oznaczeń

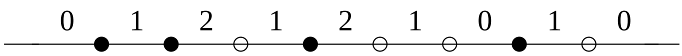
Przez $P = \{p_1, \ldots, p_n\}$ i $S = \{s_1, \ldots, s_n\}$ będziemy oznaczać zbiory punktów na płaszczyźnie, w których znajdują się odpowiednio punkty wydobycia i stacje dystrybucji gazu. Elementy zbioru P będziemy często nazywali po prostu *punktami*, zaś elementy zbioru S — *stacjami*. Wreszcie dowolny sposób połączenia stacji dystrybucji i punktów wydobycia za pomocą gazociągów południowo-wschodnich nazwiemy po prostu *przyporządkowaniem*, a przyporządkowanie nazwiemy *optymalnym*, jeżeli minimalizuje ono sumaryczną długość wykorzystanych gazociągów.

Wstępne spostrzeżenia

Z treści zadania wiemy, że punkt p i stację s można połączyć gazociągiem, jeżeli $s_x - p_x \geqslant 0$ oraz $p_y - s_y \geqslant 0$. Długość potrzebnego gazociągu wyraża się wówczas jako odległość pomiędzy p oraz s w metryce miejskiej, czyli $d(p,s) = (s_x - p_x) + (p_y - s_y)$. Zauważmy, że ta wartość jest niezależna od dokładnego przebiegu gazociągu.

Zanim poszukamy rozwiązania problemu postawionego w treści zadania, spróbujmy rozważyć jego uproszczoną wersję. Ograniczmy się mianowicie do jednowymiarowej wersji problemu, w której punkty wydobycia i stacje dystrybucji znajdują się na jednej (poziomej) prostej i zastanówmy się, jak w takiej sytuacji wygląda poszukiwane optymalne przyporządkowanie. Wymaganie, by gazociągi przebiegały w kierunku południowo-wschodnim redukuje się do wymagania, by dla każdej połączonej pary punkt–stacja o odciętych równych odpowiednio p_x i s_x zachodziła nierówność $p_x \leqslant s_x$.

Niech $[x, x+1]$, dla liczby całkowitej x, będzie odcinkiem jednostkowym na rozważanej prostej. Chcielibyśmy wiedzieć, ile gazociągów może przechodzić przez cały ten odcinek w różnych możliwych przyporządkowaniach (zauważmy, że żaden punkt ani żadna stacja nie znajdują się we wnętrzu tego odcinka). Niech P_x będzie liczbą punktów o odciętych nie większych niż x, a S_x — liczbą stacji spełniających ten warunek. Każda z tych S_x stacji musi zostać połączona z pewnym spośród wybranych P_x punktów. Ponieważ w zadaniu mamy zagwarantowane istnienie jakiegoś poprawnego przyporządkowania, to musi zachodzić $P_x \geqslant S_x$. Każdy z $P_x - S_x$ punktów wydobycia musi zostać zatem połączony z jakąś stacją dystrybucji o odciętej większej bądź równej $x+1$, co oznacza, że w *każdym* przyporządkowaniu przez odcinek $[x, x+1]$ będzie przechodzić dokładnie $P_x - S_x$ gazociągów. Zauważmy, że właśnie pokazaliśmy, że każde przyporządkowanie na prostej wymaga takiej samej łącznej długości gazociągów — jest ona równa sumie wartości $P_x - S_x$ dla wszystkich odcinków $[x, x+1]$. Innymi słowy, każde poprawne przyporządkowanie na prostej jest optymalne!



Przykładowe jednowymiarowe rozmieszczenie punktów i stacji. Dla każdego istotnego odcinka jednostkowego zaznaczony został współczynnik $P_x - S_x$.

Rozwiązanie dla przypadku jednowymiarowego pozwala nam postawić hipotezę, że również w przypadku dwuwymiarowym każde przyporządkowanie charakteryzuje się tą samą sumą długości wykorzystanych gazociągów. Okazuje się, że to stwierdzenie jest prawdziwe i na dodatek istnieje jego prosty i elegancki dowód. Suma długości gazociągów w dowolnym przyporządkowaniu wyraża się jako:

$$L = \sum_{i=1}^{n} d(p_i, s_{\pi(i)}),$$

gdzie π jest permutacją liczb od 1 do n, oznaczającą konkretne przyporządkowanie. Korzystając z definicji d oraz przemienności sumy i różnicy możemy wykonać następujące przekształcenia:

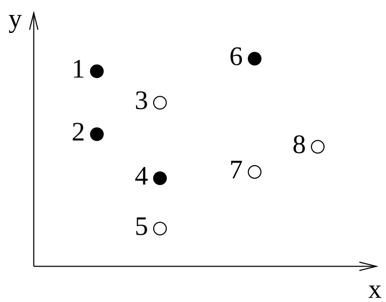
$$L = \sum_{i=1}^{n} \left(\left(s_{\pi(i)}\right)_x - (p_i)_x + (p_i)_y - \left(s_{\pi(i)}\right)_y \right) =$$

$$\sum_{i=1}^{n}\left(s_{\pi(i)}\right)_x-\sum_{i=1}^{n}(p_i)_x+\sum_{i=1}^{n}(p_i)_y-\sum_{i=1}^{n}\left(s_{\pi(i)}\right)_y=$$
$$\sum_{i=1}^{n}(s_i)_x-\sum_{i=1}^{n}(p_i)_x+\sum_{i=1}^{n}(p_i)_y-\sum_{i=1}^{n}(s_i)_y$$

Ostatnie wyrażenie pokazuje, że wartość L nie zależy od przyporządkowania, a jedynie od współrzędnych punktów i stacji. Stąd każde przyporządkowanie wymaga tej samej łącznej długości gazociągów, a zatem dowolne poprawne przyporządkowanie jest optymalne.

Rozwiązanie wzorcowe

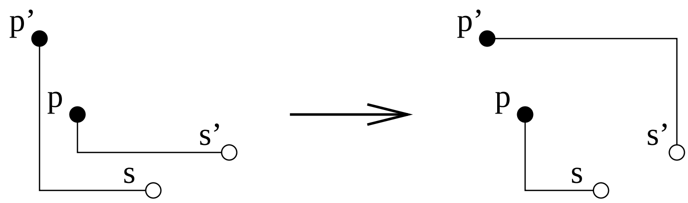
Na podstawie uprzednio poczynionych spostrzeżeń doszliśmy do wniosku, że w celu rozwiązania zadania wystarczy znaleźć dowolne *poprawne* przyporządkowanie. Wykorzystamy do tego celu technikę *zamiatania* płaszczyzny prostą pionową (*miotłą*). Na początku sortujemy wszystkie elementy (punkty i stacje) niemalejąco względem ich odciętych, rozstrzygając remisy na korzyść elementów o większej rzędnej. Następnie rozważamy elementy w otrzymanej kolejności.



Przykładowe rozmieszczenie punktów i stacji na płaszczyźnie z zaznaczoną kolejnością ich rozważania.

Jeżeli rozważany element jest punktem wydobycia, to umieszczamy go w miotle, reprezentowanej przez strukturę danych, utrzymującą punkty w porządku niemalejącym rzędnych. Z kolei przy napotkaniu stacji dystrybucji, przydzielamy jej punkt spośród znajdujących się w miotle. Zauważmy, że dzięki odpowiedniemu kryterium wstępnego sortowania, każdy punkt p znajdujący sie w miotle może zostać połączony z napotkaną stacją s, jeżeli tylko $p_y \geqslant s_y$. Udowodnimy, że jeżeli do połączenia za każdym razem wybierzemy punkt z miotły o najmniejszej rzędnej nie mniejszej niż s_y, to na końcu otrzymamy poprawne przyporządkowanie (przypomnijmy, że w zadaniu mamy zagwarantowane istnienie przynajmniej jednego poprawnego przyporządkowania).

Zastanówmy się, dlaczego postępowanie według takiego zachłannego kryterium daje poprawny wynik. Wystarczy pokazać, że zawsze istnieje poprawne przyporządkowanie, w którym pierwszą rozważaną stację s połączymy z najniżej położonym punktem miotły p, dla którego $p_y \geqslant s_y$ — jeżeli to jest prawdą, to podobne rozumowanie można przeprowadzić dla kolejno rozważanych stacji. Niech R będzie poprawnym przyporządkowaniem, w którym stacja s nie jest połączona z punktem p, lecz z innym punktem p' i niech s' będzie stacją, z którą jest połączony punkt p. Z warunków budowy gazociągów wiemy, że $s'_y \leqslant p_y$, Natomiast z faktu, że algorytm zamiatania wybiera punkt p w sposób zachłanny wynika, że $p_y \leqslant p'_y$ (p' musiał znajdować się w miotle wraz z p w momencie przydzielania punktu do stacji s, a nie został wybrany). Razem nierówności te dają zależność $s'_y \leqslant p'_y$. Na podstawie kryterium wstępnego sortowania elementów wiemy, że $s'_x \geqslant s_x$, z kolei ze sposobu prowadzenia gazociągów wynika, że $s_x \geqslant p'_x$. Łącząc te nierówności, otrzymujemy $s'_x \geqslant p'_x$. Skoro więc $s'_y \leqslant p'_y$ oraz $s'_x \geqslant p'_x$, to punkt p' i stację s' można połączyć gazociągiem. Elementy p oraz s oczywiście także można połączyć. To pozwala nam wykonać w przyporządkowaniu R odpowiednią zamianę połączeń.



Zmiana połączeń między p, p', s i s'.

Otrzymujemy w ten sposób poprawne rozwiązanie, w którym p i s są połączone, co kończy dowód poprawności algorytmu zachłannego.

Ostatnią kwestią, jaką należy rozważyć, jest wybór struktury danych, reprezentującej miotłę. Współrzędne punktów możemy przechowywać w zwykłej tablicy i za każdym razem wybierać (w czasie liniowym względem liczby punktów w miotle, a więc pesymistycznie liniowym względem n) punkt spełniający kryterium zachłanne. Otrzymamy w ten sposób rozwiązanie o złożoności czasowej $O(n^2)$, zaimplementowane w pliku `gazs0.c`. Takie rozwiązanie uzyskiwało 40% punktów możliwych do zdobycia za zadanie. Można także spróbować wykorzystać strukturę w postaci zrównoważonego, binarnego drzewa poszukiwań, w którym w złożoności czasowej $O(\log n)$ można wyznaczyć dla danej wartości punkt o najmniejszej rzędnej nie mniejszej od niej; dzięki temu złożoność czasowa całego rozwiązania wyniesie $O(n \log n)$. Programując w C++, można sięgnąć po STL-owy kontener `set`, w którym dysponujemy odpowiednią metodą `lower_bound()` (implementacja rozwiązania z tą strukturą znajduje się w pliku `gaz.cpp`). Inną prostą w implementacji strukturą, którą można zastosować, jest statyczne drzewo licznikowe (skonstruowane na samym początku dla całego zakresu rzędnych). W strukturze tej w każdym węźle utrzymujemy liczbę punktów zawartych w poddrzewie, którego jest on korzeniem. Za pomocą jednego przejścia od korzenia drzewa do odpowiedniego liścia możemy zarówno wstawić nowy punkt do struktury, jak i wykonać zapytanie o punkt o najmniejszej rzędnej nie mniejszej od zadanej. Na podstawie ograniczeń z zadania możemy przyjąć, że maksymalna możliwa rzędna punktu jest rzędu $O(n)$, co pozwala oszacować złożoność czasową każdej z wyżej wymienionych operacji jako $O(\log n)$. Ponieważ drzewa licznikowe są ogólnie znane, to dokładny opis budowy oraz implementacji operacji na tej wersji struktury miotły pozostawiamy Czytelnikowi jako ćwiczenie. Implementacja rozwiązania z drzewem licznikowym znajduje się w plikach `gaz0.c` i `gaz1.pas`. Każde z rozwiązań wzorcowych ma złożoność pamięciową $O(n)$.

Inne rozwiązania

Zauważmy, że polecenie z zadania można przeformułować na poszukiwanie w grafie dwudzielnym (jedną grupą wierzchołków są punkty, drugą stacje, a krawędź istnieje wtedy, kiedy da się poprowadzić gazociąg) najtańszego doskonałego skojarzenia. Używając najlepszego implementowalnego w praktyce algorytmu, rozwiązującego ten problem (np. metody węgierskiej lub Busackera-Gowena z wykorzystaniem algorytmu Dijkstry — patrz opis rozwiązania zadania Szkoły z XIII Olimpiady Informatycznej), da się skonstruować rozwiązanie o złożoności czasowej $O(n^3)$. Dzięki spostrzeżeniu, że każde przyporządkowanie jest optymalne, można też zamiast najtańszego szukać po prostu dowolnego doskonałego skojarzenia w grafie. W pliku `gazs1.c` znajduje się rozwiązanie tego problemu, wykorzystujące metodę Edmondsa-Karpa (złożoność czasowa $O(n^3)$). Używając szybszych metod, na przykład algorytmu Hopcrofta-Karpa, można by otrzymać rozwiązanie o złożoności czasowej nawet $O(n^2\sqrt{n})$. Widać jednak wyraźnie, że sprowadzając nasz problem do wyznaczania skojarzenia w grafie dwudzielnym nie możemy osiągnąć zbyt efektywnych algorytmów, jako że samych krawędzi w tym grafie jest $O(n^2)$. Stwarza to także problemy w zakresie złożoności pamięciowej: przechowywanie wszystkich krawędzi w pamięci jest niemożliwe do zrealizowania dla dużych danych wejściowych, natomiast wyznaczanie krawędzi online jest z kolei czasochłonne. Rozwiązania tego typu nie uzyskiwały zazwyczaj więcej niż 30% punktów.

Najczęstszy błąd popełniany przez zawodników polegał na zapomnieniu o konieczności użycia typów całkowitych 64-bitowych przy wyznaczaniu sumy długości gazociągów w optymalnym przyporządkowaniu, co powodowało utratę 20% punktów.

Testy

Zadanie było sprawdzane na 10 zestawach testów. Większość testów została wygenerowana w sposób losowy. Dokładniejszą ich charakterystykę przedstawia poniższa tabelka:

Nazwa	n	Opis
gaz1a.in	6	test wygenerowany ręcznie
gaz1b.in	50	test wygenerowany ręcznie
gaz2a.in	100	test losowy gęsty
gaz2b.in	100	test losowy rzadki
gaz3a.in	100	test losowy gęsty
gaz3b.in	100	test losowy rzadki
gaz4a.in	1 000	test losowy gęsty
gaz4b.in	1 000	test losowy rzadki
gaz5a.in	30 000	test losowy gęsty
gaz5b.in	30 000	test losowy rzadki
gaz6a.in	50 000	test losowy gęsty

Nazwa	n	Opis
gaz6b.in	50 000	test losowy rzadki
gaz7a.in	40 000	test losowy gęsty
gaz7b.in	40 000	test losowy rzadki
gaz8a.in	40 000	test losowy gęsty
gaz8b.in	40 000	test losowy rzadki
gaz9a.in	40 000	test z dużym wynikiem
gaz9b.in	50 000	test losowy rzadki
gaz10a.in	50 000	test z dużym wynikiem
gaz10b.in	50 000	test losowy rzadki
gaz10c.in	50 000	wszystkie punkty leżą na jednej prostej i są posortowane

W testach gęstych wszystkie punkty i stacje znajdują się w jednym niedużym kwadracie, natomiast w testach rzadkich są one rozmieszczone na dużo większym obszarze.