

Zadanie G2 - Firma kurierska: relaksacja ograniczenia dolnego 1-tree i zachłanne ograniczenie górne

Podczas rozwiązywania problemu komiwożacza naszym głównym celem jest znalezienie dobrej trasy. W celu oceny jakości uzyskanych rozwiązań potrzebujemy dobrego ograniczenia dolnego na długość optymalnej trasy.

Znajdź ograniczenie dolne dla problemu komiwożacza wykorzystując schemat relaksacji Lagrange'a dla ograniczenia 1-tree. Dodatkowo wyznacz trasę zbudowaną zachłannie na podstawie 1-drzewa świadczącego znalezione ograniczenie dolne.

Wejście

Pierwsza linia wejścia zawiera liczbę całkowitą z ($1 \leq z \leq 2 \cdot 10^9$) – liczbę zestawów danych, których opisy występują kolejno po sobie. Opis jednego zestawu jest następujący:

W pierwszej linii zestawu znajduje się liczba naturalna n oznaczająca liczbę miast opisanych jako punkty na płaszczyźnie ($2 \leq n \leq 10000$). W kolejnych n liniach znajdują się współrzędne punktów, tzn. dwie liczby rzeczywiste a i b . Są one podane zgodnie z kolejnością występowanie na trasie komiwożacza wyznaczonej przez *heurezę Christofidesa*.

Wyjście

Dla każdego zestawu danych wypisz: w pierwszej linii sufit z wagi 1-drzewa w zmodyfikowanej instancji, czyli znalezione ograniczenie dolne na długość optymalnej trasy w podanym zestawie. W drugiej linii wypisz trasę komiwożacza zbudowaną zachłanną *heurezą* z 1-drzewa świadczącego ograniczenie dolne, w trzeciej linii wypisz koszt uzyskanej trasy.

Dostępna pamięć: 850MB

Uwagi o rozwiązaniu wzorcowym

- Grafy kandydatów wyznaczane w programie są sumą triangulacji Delaunay'a oraz 10-nearest neighbours.
- 500 iteracji relaksacji.
- krok początkowy równy jest $step = 10 \cdot \frac{upper - lower}{n}$, gdzie *upper* to długość trasy obliczonej przez *heurezę Christofides* (podanej na wejściu), *lower* to ograniczenie dolne na długość optymalnej trasy obliczone metodą 1-tree, n to liczba wierzchołków w instancji.
- w kolejnych iteracjach krok zmniejszany jest za pomocą formuły $step = step \cdot LAMBDA$, gdzie $LAMBDA = 0.98$.
- wagi na wierzchołkach na początku ustawione są na zero; w kolejnych iteracjach wagi wierzchołków obliczane są wg wzoru $p[i] = p[i] + step \cdot (0.7 \cdot d[currentD][i] + 0.3 \cdot d[1 - currentD][i])$, gdzie $p[i]$ przed instrukcją było dotychczasową wagą i-tego wierzchołka, $d[currentD][i]$ to stopień i-tego wierzchołka w 1-drzewie znalezionym dla obecnej, zmodyfikowanej instancji pomniejszony o 2 (czyli odległość stopnia od oczekiwanego stopnia w trasie), $d[1 - currentD][i]$ to stopień i-tego wierzchołka w 1-drzewie znalezionym dla poprzedniej (krok wcześniej) instancji pomniejszony o 2.

- 1-drzewo obliczane dla każdej zmodyfikowanej instancji jest liczone w przybliżeniu na grafie kandydatów. Najpierw obliczamy minimalne drzewo rozpinające T w grafie kandydatów, a później szukamy krawędzi $e = (u, v)$ realizującej \max_u liść $\min_{v \neq u, v \neq \text{parent}(u)} c'(u, v)$.
- po wykonaniu wszystkich iteracji wybieramy najcięższe 1-drzewo (najlepsze ograniczenie dolne dla trasy w grafie kandydatów) i dla dokładnie tych samych wag $p[i]$ dla których to przybliżenie było obliczone liczymy poprawnie 1-drzewo. Zatem obliczamy minimalne drzewo rozpinające całego grafu (rozwiązanie wzorcowe używa algorytmu Kruskala i jest to krytyczna część programu dla wielkości wymaganej pamięci) i dokładamy do tego drzewa najtańszą znaną krawędź niedrzewową o choć jednym końcu w liściu drzewa.
- $\lceil \text{waga 1-drzewa} - 2 \cdot \sum p[i] \rceil$ jest ograniczeniem dolnym na długość trasy w rozważanej instancji.
- trasa komiwożacza obliczana jest zachłannie z konstruowanych 1-drzew. Podczas 150 ostatnich iteracji algorytm konstruuje trasę z aktualnych 1-drzew i wybiera najlepszą znaną trasę jako ostateczną odpowiedź.

Uwagi ogólne

- Odległość dwóch punktów obliczana jest za pomocą następującej funkcji:

```
vector<Delaunay::Point> Points;  
inline int euclid(int v, int w) {  
    double x = Points[v].x() - Points[w].x();  
    double y = Points[v].y() - Points[w].y();  
    return floor(sqrt(x*x+y*y) + 0.5);  
}
```

- Jakość ograniczenia dolnego mierzona jest względnym odchyleniem od długości trasy optymalnej (długości te znane są dla testów tsplib) lub od wartości ograniczenia górnego wyznaczonego metodą *Lin-Kernighan*). Jakość trasy w jednym pliku wejściowym jest uśredniana. Dopuszczalne są poprawne rozwiązania o średniej jakości co najwyżej 1 – 1.5% gorszej od jakości rozwiązań wzorcowych. Progi dla kolejnych plików wejściowych prezentuje poniższa tabelka (pierwsza wartość).
- Jakość ograniczenia górnego mierzona jest na podstawie różnicy między otrzymanym ograniczeniem górnym i dolnym. Dopuszczalne są poprawne rozwiązania o średniej jakości co najwyżej 2.5 – 3.5% gorszej od jakości rozwiązań wzorcowych. Progi dla kolejnych plików wejściowych prezentuje poniższa tabelka (druga wartość).

Pliki wejściowe:	Progi:
rel-0	8.5 15.5
rel-1	6.5 9.5
rel-2	5.5 11.5
rel-3	5 13
rel-4	5 13
rel-5	18.5 24.5
rel-tsplib	4.5 17

Przykład

Dla danych wejściowych:	Poprawną odpowiedzią jest:
2	6
6	0 1 2 3 4 5
0 0	6
1 0	40
2 1	0 1 2 3 5 4 6 7 15 14 13 12 11 10 8 9
1 1	50
0.5 0.5	
0 1	
16	
0 0	
1 0	
3 0	
5 0	
3 5	
5 5	
1 5	
0 5	
1 15	
0 15	
3 15	
5 15	
5 10	
3 10	
1 10	
0 10	