

Roy-Floyd. Ciclu eulerian

BUZATU GIULIAN & NIȚĂ ALEXANDROS

Problemă

Dându-se un graf ponderat, aflați distanțele minime pentru fiecare pereche de noduri.

Problemă

Dându-se un graf ponderat, aflați distanțele minime pentru fiecare pereche de noduri.

Cum am putea face?

Problemă

Dându-se un graf ponderat, aflați distanțele minime pentru fiecare pereche de noduri.

Putem implementa algoritmul lui Dijkstra, dacă graful are doar ponderi pozitive, sau algoritmul Bellman-Ford, dacă avem și costuri negative, și după să rulăm pe rând algoritmul pentru fiecare nod. Totuși, acest lucru ne-ar genera o complexitate destul de mare.

Roy-Floyd

Pentru a rezolva problema anterioară, putem folosi algoritmul Roy-Floyd. Acesta are complexitatea $O(n^3)$ și acceptă costuri negative, dar nu și cicluri negative. Acest algoritm este unul de programare dinamică.

Roy-Floyd

Avem o matrice $d[i][j]$ ce reprezintă distanța minimă dintre nodurile i și j . Inițial, matricea va conține doar muchiile directe dintre i și j , 0 pe diagonala principală și ∞ în rest.

Pentru fiecare nod k , considerăm drumurile pe care le putem augmenta folosind nodul k . Practic, la pasul k , noi vom avea deja calculate drumurile de cost minim folosind doar nodurile din mulțimea $\{1, 2, \dots, k-1\}$ și vom folosi nodul k pentru a vedea dacă putem îmbunătăți distanțele deja obținute. Deci, folosim k pentru a uni lanțul de la i la k și lanțul de la k la j , astfel:

$$d[i][j] = \min(d[i][j], d[i][k] + d[k][j])$$

Roy-Floyd

Problemă: <https://infoarena.ro/problema/royfloyd>

Implementare: https://infoarena.ro/job_detail/3193701?action=view-source

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/coach>

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/coach>

Hint: Ne-ar ajuta să avem nodurile sortate crescător după valoarea lor calorică.

Problemă

Să rezolvăm problema: <https://infoarena.ro/problema/coach>

Soluție: Sortăm nodurile după valoarea calorică și le renumerotăm. Acum putem fixa valoarea calorică minimă și să aplicăm algoritmul Roy-Floyd. După fiecare pas k verificăm dacă am atins cumva valoarea dorită T . Dacă da, atunci reținem valorile care ne-au oferit răspunsul și putem să ne oprim din căutare. Deoarece noi am obținut răspunsul folosind noduri din mulțimea $\{1, 2, \dots, k\}$, valoarea calorică maximă este cea a nodului k .

Implementare: https://infoarena.ro/job_detail/3193753?action=view-source

Graf eulerian

Un lanț eulerian este un lanț în care apare fix o dată fiecare muchie din graf. Similar definim noțiunea de ciclu eulerian.

Graf eulerian

Un lanț eulerian este un lanț în care apare fix o dată fiecare muchie din graf. Similar definim noțiunea de ciclu eulerian.

Dându-se un multigraf (un graf în care pot exista muchii de la un nod la el însuși și pot exista mai multe muchii între două noduri), aflați dacă acesta este sau nu un graf eulerian (i. e. conține un ciclu eulerian). În caz afirmativ, afișați ciclul eulerian.

Ciclu eulerian

Teoremă: Un multigraf este eulerian dacă:

1. este conex
2. gradul fiecărui nod este par

Ciclu eulerian

Teoremă: Un multigraf este eulerian dacă:

1. este conex
2. gradul fiecărui nod este par

Această teoremă rezolvă prima parte a problemei. Dar cum putem să aflăm ciclul, dacă multigraful nostru este eulerian?

Ciclu eulerian

Teoremă: Un multigraf este eulerian dacă:

1. este conex
2. gradul fiecărui nod este par

Această teoremă rezolvă prima parte a problemei. Dar cum putem să aflăm ciclul, dacă multigraful nostru este eulerian?

Hint: Gândiți-vă la cum arată un ciclu eulerian.

Ciclu eulerian

Algoritmul se bazează pe observația că un ciclu eulerian se poate descompune într-o reuniune de cicluri disjuncte. Astfel, algoritmul va parcurge fiecare ciclu disjunct și va memora într-un vector această parcurgere. Vom face acest lucru recursiv, într-un mod similar cu o parcurgere DFS.

Ciclu eulerian

Descriere algoritm:

1. Pornim dintr-un nod de start u .
2. Mergem într-un vecin nevizitat al nodului de start și ștergem muchia care îi unește.
3. Pornim o altă căutare a unui ciclu eulerian din nodul vizitat.
4. Când terminăm de parcurs ciclul, îl memorăm invers.

Complexitate: $O(n + m)$ - algoritmul fiind similar cu un DFS.

Implementare: https://infoarena.ro/job_detail/3194826?action=view-source

Problemă

Să rezolvăm problema: <https://codeforces.com/gym/104013/problem/E>

Problemă

Să rezolvăm problema: <https://codeforces.com/gym/104013/problem/E>

Hint 1: Cum am rezolva problema dacă am avea doar operații pe care trebuie să le facem?

Problemă

Să rezolvăm problema: <https://codeforces.com/gym/104013/problem/E>

Hint 1: Cum am rezolva problema dacă am avea doar operații pe care trebuie să le facem?

Hint 2: Cum putem integra operațiile care nu trebuie făcute în soluția precedentă?

Problemă

Să rezolvăm problema: <https://codeforces.com/gym/104013/problem/E>

Hint 1: Cum am rezolva problema dacă am avea doar operații pe care trebuie să le facem?

Hint 2: Cum putem integra operațiile care nu trebuie făcute în soluția precedentă?

Hint 3: Există două tipuri de operații pe care nu trebuie să le facem.

Problemă

Să rezolvăm problema: <https://codeforces.com/gym/104013/problem/E>

Soluție: Putem construi un graf unde punem muchie de la nodul a la nodul b , dacă avem o operație de la a la b cu $w=1$. Putem găsi un lanț/ciclu eulerian în acest graf. Dacă avem unul, atunci știm că putem face toate operațiile cu $w=1$. Trebuie să vedem dacă putem integra operațiile cu $w=0$ în soluția curentă. Putem observa că sunt două tipuri de operații cu $w=0$, cele care încep din nodul de start c și celelalte. Așadar, dacă putem integra aceste operații în lanțul/ciclul nostru eulerian, avem soluție. Din cauză că datele de intrare sunt mari, trebuie să le normalizăm la început.

Implementare: <https://github.com/Giulian617/Hai-la-olimpiada-2023-2024/blob/main/11-12/resources/GYM104013E.cpp>

Temă

- <https://infoarena.ro/problema/rf>
- <https://infoarena.ro/problema/rfinv>
- <https://infoarena.ro/problema/domino>
- <https://infoarena.ro/problema/cartite>

Probleme suplimentare

- <https://infoarena.ro/problema/tester>
- <https://infoarena.ro/problema/ulei>
- <https://infoarena.ro/problema/mmo>

Lectură suplimentară

- <https://cp-algorithms.com/graph/all-pair-shortest-path-floyd-warshall.html>
- <https://arxiv.org/pdf/1904.01210.pdf>
- https://cp-algorithms.com/graph/euler_path.html