

Főváros

Innen: Algowiki

Tartalomjegyzék

- 1 Feladat
 - 1.1 Az eredeti feladat
- 2 Megoldási ötletek
 - 2.1 Hibás megoldási ötletek és ellenpéldák
 - 2.2 Helyes, de lassú megoldások
- 3 Segítségek
- 4 Megoldás
 - 4.1 Fontos gondolatok
 - 4.2 Részletes megoldás
 - 4.3 Helyesség indoklása
- 5 Komplexitás
- 6 Implementáció

Feladat

N darab várost M db egyirányú út köt össze. Egyet ki szeretnénk nevezni fővárosnak, de biztonsági okokból a fővárost bárhonnan el kell tudni érni. Mely városok érhetőek el az összes többiből? Feltehetjük, hogy létezik ilyen város.

Az eredeti feladat

Sphere Online Judge (<https://www.spoj.com/problems/CAPCITY/>)

Megoldási ötletek

Hibás megoldási ötletek és ellenpéldák

Nem tudunk egyszerűen topologikusan rendezni, mert a gráfban lehetnek körök.

Helyes, de lassú megoldások

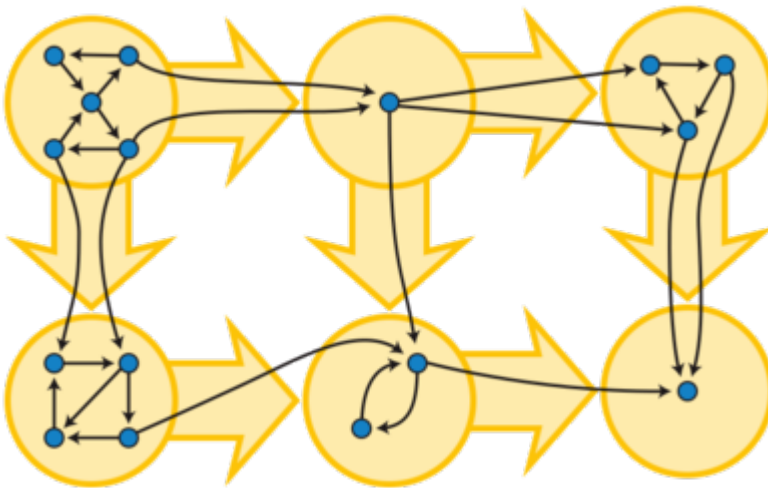
A kapott utakat fevehetnénk ellenkező irányban, majd indíthatnánk minden egyes városból gráfbejárást, megjelölve azokat, ahonnan elértünk minden várost. Ez legjobb esetben is $O(|V|(|V| + |E|))$, azaz négyzetes időbeli komplexitású, tehát sok város esetén hamar túl lassúvá válik.

Segítségek

A topologikus rendezés ígéretesnek hangzik, mert egy irányított körmentes gráfról könnyen és gyorsan megállapíthatjuk vele a pontok egymástól való függését. Egyszerűen megnézzük, hogy melyik az utolsó elem. Viszont nekünk nem körmentes a gráfunk, és nem csak egyetlen elemre vagyunk kíváncsiak. Hogyan alakíthatjuk át a gráfot úgy, hogy topologikus rendezéssel megoldható legyen a probléma?

Megoldás

Fontos gondolatok



Egy irányított gráfban kialakíthatók olyan csoportok, melyeken belül minden pont elérhető minden másik pontból. Ezeket Erősen egybefüggő komponenseknek (https://hu.wikipedia.org/wiki/Er%C5%91sen_%C3%B6sszef%C3%BCgg%C5%91_komponens) hívjuk. Ezek az irányított gráfban egyetlen pontnak is tekinthetők, ahol a bemenő és kimenő élek a csoport pontjainak bemenő és kimenő éleinek összessége.

Ha minden ilyen csoportot megtalálunk és a csoportok pontjait összevonjuk, egy körmentes gráfot kapunk. A probléma eredeti oldalán az #scc tag is erre a koncepcióra utal.

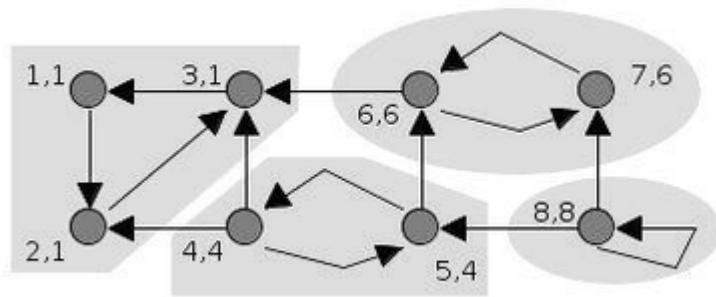
Ezen az új gráfon már végezhetünk topologikus rendezést, és az utolsó helyre került csoport elemei lesznek a lehetséges fővárosok.

Részletes megoldás

Mivel a négyzetes megoldás lassú, lineáris időben kell megoldani a feladatot, azaz fix mennyiségű bejárással. Meg kell határozni az erősen összefüggő komponenseket, például Tarjan (https://en.wikipedia.org/wiki/Tarjan%27s_strongly_connected_components_algorithm) vagy Kosaraju (https://en.wikipedia.org/wiki/Kosaraju%27s_algorithm) algoritmusával, majd azok függését egymástól

topologikus sorba rendezéssel (https://en.wikipedia.org/wiki/Topological_sorting). A topologikus sor utolsó csoportja tartalmazza a lehetséges fővárosokat.

Ha Tarjan algoritmusát (https://en.wikipedia.org/wiki/Tarjan%27s_strongly_connected_components_algorithm) alkalmazva csoportosítjuk a pontokat a gráf erősen egybefüggő komponenseivé, az algoritmus nemcsak hogy lineáris időben, ráadásul fordított topologikus sorrendben találja meg a komponenseket, így az első megtalált komponens tagjai a lehetséges fővárosok.



Helyesség indoklása

A topologikusan rendezett csoportok utolsó eleme definíció alapján bármely korábbi csoportból elérhetőek (mert az input gráfnak legfeljebb egyetlen komponense van), és az utolsó csoport elemei definíció alapján elérhetőek egymásból, tehát bármelyikük lehet főváros.

Komplexitás

Mivel egyetlen mélységi bejárást végzünk így az időbeli komplexitás lineáris $O(|V| + |E|)$, és mivel minden ponthoz két segédváltozót rendelünk, így a memóriabeli komplexitás is az: $O(|V|)$

Implementáció

SPOJ által elfogadott implementáció a repl.it-en (<https://repl.it/@szabin/SPOJ-CAPCITY>)

A lap eredeti címe: „<https://algowiki.miraheze.org/w/index.php?title=Főváros&oldid=1245>”