Checkposts

Innen: Algowiki

Tartalomjegyzék

Feladat

Egy városban N darab kereszteződés ($1 \leq N \leq 10^5$), és M darab egyirányú út van ($0 \leq M \leq 3 \cdot 10^5$). A polgármester biztonságossá akarja tenni a várost rendőri ellenőrzőpontok elhelyezésével. Egy ilyen ellenőrzőpontok csak kereszteződésben lehetnek. Ha az i. kereszteződésbe helyezünk egy ellenőrzőpontot, akkor az biztonságossá tesz minden olyan j. kereszteződést, ahol i = j, vagy i-ből el lehet jutni j-be és vissza. Ezen felül különböző kereszteződésekbe különböző költséggel jár egy ellenőrzőpont építése (semelyik ilyen költség nem haladja meg a 10^9-t).

Cél: Meghatározni a város (azaz az összes kereszteződés) biztonságossá tételének **minimális költségét**, továbbá meghatározni annak a számát, hogy **hányféleképpen** tudjuk ezt megtenni úgy, hogy a lehető **legkevesebb ellenőrzőpontot építünk** (és az árat is minimumon tartjuk).

Az eredeti feladat

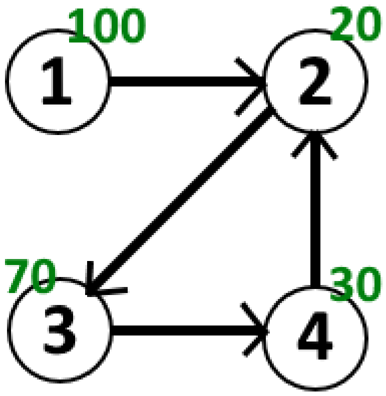
Az eredeti feladat megtekinthető itt (https://codeforces.com/contest/427/problem/C), a Codeforces-on.

Megoldási ötletek

Hibás megoldási ötletek és ellenpéldák

Ötlet: Rendezzük a kereszteződéseket növekvő sorrendbe aszerint, hogy mennyibe kerül oda ellenőrzőpontot építeni. Majd elkezdünk építeni ellenőrzőpontokat a legolcsóbb helyekre, míg nem lesz teljes lefedettségünk.

Ellenpélda: Nézzük a mellékelt ábrát. A a számozott körök a kereszteződések, a nyilak az egyirányú utak, a zöld számok az építési költségek.



Ha belegondolunk az 1-es kereszteződésbe muszáj ellenőrzőpontot építeni, mert ezt semelyik másik nem fedné le (semelyik másikból nem lehet az 1-esbe eljutni). Ugye az 1-es építési költsége a legdrágább, tehát ha elkezdenénk a legolcsóbbakba építgetni, akkor itt az összesbe építenünk kellene, mire teljes lenne a lefedettség. Azonban itt bőven elegendő lenne, az 1-esbe és a 2-esbe építeni a teljes lefedettséghez.

Helyes, de lassú megoldások

Nézzünk inkább egy biztos megoldást. Próbáljuk ki az összes lehetséges építést, nézzük meg melyikek jók, és azok közül mennyi a minimum ár. Viszont ez egy exponenciális, azaz $\mathbf{O(2^N)}$-es műveletigénnyel járna, mivel minden egyes kereszteződést a többitől függetlenül ki kellene próbálnunk úgy, hogy építünk oda ellenőrzőpontot és úgy hogy nem. A feladat korlátait ismerve sajnos ez túl lassú megoldás.

Segítségek

1. segítség: Goldoljunk úgy a városra, mint egy irányított gráfra.

2. segítség: Mit takar ez a lefedettség ebben a gráfban? Azaz mit jelent az, hogy i-ből el tudunk menni j-be és vissza?

Megoldás

Fontos gondolatok

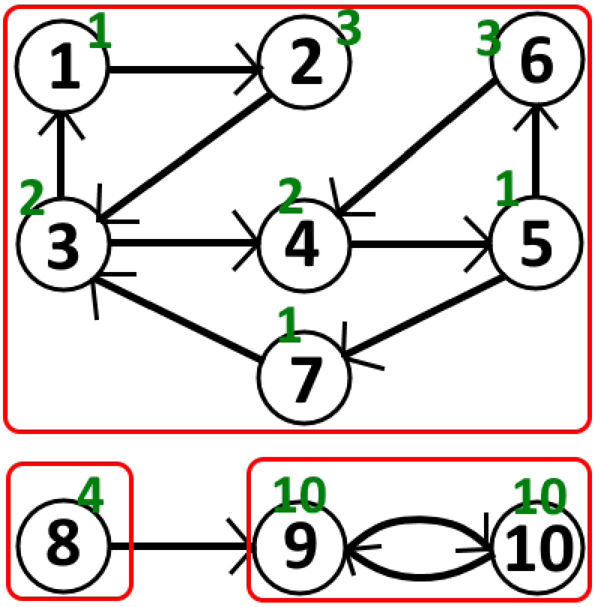
A feladat szövege átgondolva: Van egy N csúcsú, M élt tartalmazó irányított gráfunk. Ki kell választanunk csúcsokat, úgy hogy minden erősen összefüggő komponensből válasszunk legalább egyet.

Részletes megoldás

Az erősen összefüggő komponensek meghatározására használhatjuk a Kosaraju algoritmust, amivel eltárolhatjuk a komponenseket. Ugye minimalizálni kell a költséget, tehát minden komponensből a legalacsonyabb költségűt kell kiválasztani. Majd összeadjuk minden komponens legolcsóbbjának árát és kész is a megoldás első része.

Még meg kell határoznunk azt, hogy minimális költséggel és ellenőrzőponttal hányféleképpen tudjuk a teljes lefedést kivitelezni. Az ellenőrző pontok száma akkor lesz minimális, ha minden komponensből csak egyet választunk. A minimális költségért minden komponensből a legolcsóbbat kell választanunk. Ahhoz, hogy megkapjuk hányféleképpen tudjuk ezt kivitelezni, minden komponensben meg kell számolni hány legolcsóbb csúcs van, majd ezeket összeszorozni.

Példa



A feladat 3. példabemenetét ábrázoltam irányított gráfként. Látszik 3 erősen összefüggő komponensünk van. Ha mindegyikből összeadjuk a legolcsóbbakat, akkor kijön az **1 + 4 + 10 = 15** (azaz a minimum ár). Ha mindegyikből összeszorozzuk a minimumok számát, akkor kijön **3 * 1 * 2 = 6** (azaz az, hogy hányféleképpen tudjuk megtenni ...)

Komplexitás

A kosaraju műveleteigénye **O(N + M)** éllistás gráfreprezentációval. A komponenseken belüli minimum keresések összegzett műveletigénye **O(N)**, mert minden csúcson csak egyszer megyünk át (mert csak 1 komponenshez tartozik). Így az összegzett műveletigén **O(N + M)**. Hasonlóan a tárigény is.

Implementáció

Az **C++**-os implementációt megnézhetitek itt (https://pastebin.com/qErwUUD2). Megjegyezném, hogy a Kosaraju-hoz szükséges a gráf transzponáltja. A program gyorsítása végett én ezt is eltároltam. M

A lap eredeti címe: „https://algowiki.miraheze.org/w/index.php?title=Checkposts&oldid=1326”