Adott ponton átmenő legrövidebb kör

Innen: Algowiki

Tartalomjegyzék

Feladat

Keressük meg egy irányítatlan gráfban egy adott ***P*** ponton átmenő legrövidebb kört!

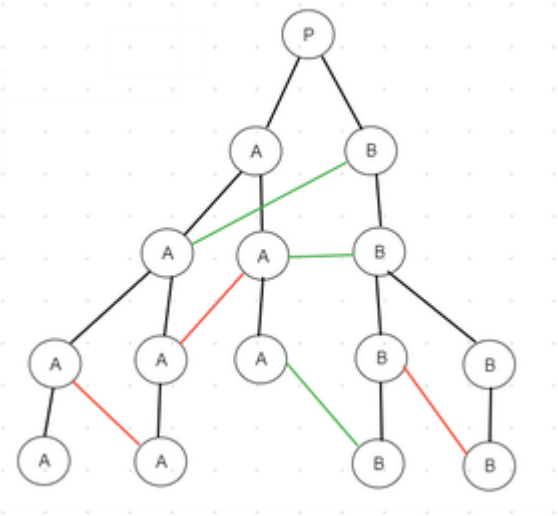
Az eredeti feladat

Az eredeti feladat elérhető a mesteren (https://mester.inf.elte.hu/): Haladó / Gráfok, suélességi bejárás / 5. feladat

Megoldás

Legyenek ***P*** szomszédjai a Q_i pontok.

Ezekből a pontokból indulva futtassunk szélességi bejárást, és képezzünk Q_i gyökerű részfákat úgy, hogy eltároljuk minden pontra (a ***P***-től vett távolsága mellett), hogy melyik pont a gyökere (az ábrán *A* és *B*).



Ha olyan, már látogatott ponthoz érünk, amelyik másik részfában van (az ábrán zöld élen keresztül), akkor egy olyan kört találtunk, ami átmegy ***P***-n *(az ábrán piros éllel jelölt körök nem mennek keresztül **P**-n).* Ilyenkor ennek a körnek a hossza a két részfát összekötő él végpontjainak távolsága ***P***-től +1 *(az őket összekötő él).*

Az ilyen köröknek kell venni a minimumát.

Ha ki is kell írni a legrövidebb körön lévő pontokat, akkor tároljuk el a pontok szülőjét is. Így a két részfát összekötő éltől indulva, két irányban, ki lehet írni a pontokat ***P***-ig.

Komplexitás

O(N+M)*, ahol N a csúcsok száma, M az élek száma.*

Implementáció

[becsuk]

C++ nyelven, a mesteres feladat szerint

```
#include <iostream>
#include <vector>
#include <queue>
using namespace std;

int minimum, N, M, P;
int main()
{
  cin >> N >> M >> P;
  minimum = N + 1; //minimum distance from P
  vector<vector<int>> G(N + 1);
  vector<bool> visited(N + 1, false);
  vector<int> parent(N + 1);
  vector<int> start(N + 1);       //starting point of the subtree
  vector<int> dist(N + 1, N + 1); //distance from P
  int x, y;
  for (int i = 0; i < M; i++) {
    cin >> x >> y;
    G[x].push_back(y);
    G[y].push_back(x);
  }
  int a = -1;
  int b = -1;
  queue<int> q;
  visited[P] = true;
  dist[P] = 0;
  for (int v : G[P]) //starting point of the subtrees
  {
    q.push(v);
    visited[v] = true;
    dist[v] = 1;
    start[v] = v;
    parent[v] = P;
  }
  //dfs
  while (!q.empty()){
    int current = q.front();
    q.pop();
    for (int v : G[current]){ //current's neighbours
      if (!visited[v]){
        q.push(v);
        visited[v] = true;
        parent[v] = current;
        dist[v] = dist[current] + 1;
        start[v] = start[current];
      }
      else if (start[v] != start[current] && v != P) { //found a circle
        if (dist[v] + dist[current] + 1 < minimum) { //found a smaller circle
          //the endpoints of the edge that connects the subtrees:
          a = v;
          b = current;
          minimum = dist[a] + dist[b] + 1;
        }
      }
    }
  }

  if (a != -1){ //found circle
    cout << minimum << endl;
    cout << P << ' ';
    //print route from P to B (need to reverse)
    vector<int> from_b;
    while (b != P) {
      from_b.push_back(b);
      b = parent[b];
    }
    for (int i = from_b.size(); i > 0; i--){
      cout << from_b[i - 1] << ' ';
```

```
    }
    //print route from A to P
    while (a != P){
      cout << a << ' ';
      a = parent[a];
    }
    cout << endl;
  }

  else{ // valid circle does not exist
    cout << -1 << endl;
  }
  return 0;
}
```

A lap eredeti címe: „https://algowiki.miraheze.org/w/index.php?title=Adott_ponton_átmenő_legrövidebb_kör&oldid=1365”