

Codeforces 580C Kefa and Park

Innen: Algowiki

Tartalomjegyzék

- 1 Feladat
 - 1.1 Eredeti feladat
- 2 Megoldás
 - 2.1 Ötlet
 - 2.2 Részletes megoldás
- 3 Komplexitás
- 4 Implementáció
 - 4.1 Kód

Feladat

Egy N csúcsú faábrában minden csúcshoz hozzá van rendelve egy logikai érték, miszerint van-e rajta macska, vagy nincs. Adjuk meg hány olyan levelet tudunk elérni a gyökérből indulva, amihez legfeljebb M egymást követő macskát tartalmazó csúcson kell átmenni.

Eredeti feladat

Codeforces - 580C Kefa and Park: <https://codeforces.com/problemset/problem/580/C>

Megoldás

Ötlet

Ha a pontokat egy structként kezeljük, aminek egyik tagja egy logikai érték (van-e macska vagy nincs), másik pedig a csúcs szomszédsági listája, akkor egy egyedi mélységi bejárással végig tudunk menni minden olyan csúcson amit a feladat leírása szerint el tudunk érni.

Részletes megoldás

A mélységi bejárás esetében a csúcs számát és az éppen vizsgált csúcsba érkezve elhagyott egymást követő macskák számát leszámítva minden paramétert kezelhetünk globális változóként, de ezt most mellőzzük, és átadjuk a függvénynek. Használunk egy Node vektort, ami a gráfot fogja reprezentálni (ahol a Node a csúcs structja), egy logikai értékek vektorát, amiben a bejárt csúcsokat tudjuk vizsgálni, a megengedett maximum macskák számát, valamint a jelenlegi macskák számát. Ha túlléptük a maximumot, akkor azt a Node-ot már nem járjuk be, és annál mélyebbre sem megyünk az ágon. Ha van macska a csúcsunkon, akkor meghívjuk a bejárást 1-el inkrementálva a macskák számát, ha nincs, akkor lenullázva hívjuk meg azt. A végén a bool vektorunkon végigfutva megvizsgáljuk minden csúcsról, hogy levél volt-e, és hogy bejártuk-e. Végül kiírjuk azok darabszámát, amelyekre mindkettő teljesült.

Komplexitás

A mélységi bejárás időkomplexitása legrosszabb esetben $O(E + V)$, ahol E az élek, és V a csúcsok száma. Jelen helyzetben ezek egymástól függenek, fagráf definíciójából adódóan $E = V - 1$. A megfelelő csúcsok összeszámlálása lineáris $O(N)$.

Implementáció

C++

Kód [becsuk]

Kód:

```

1 #include <iostream>
2 #include <vector>
3
4 struct Node
5 {
6     bool cat = false;
7     std::vector<int> connections;
8 };
9
10 void DFS_cats(int start, const std::vector<Node>& Graph, std::vector<bool>& visited, const int consecutive_cats,
11 const int allowed)
12 {
13     if (consecutive_cats <= allowed)
14     {
15         visited[start] = true;
16         for (int v : Graph[start].connections)
17         {
18             if (!visited[v])
19             {
20                 if (Graph[v].cat)
21                     DFS_cats(v, Graph, visited, consecutive_cats + 1, allowed);
22                 else
23                     DFS_cats(v, Graph, visited, 0, allowed);
24             }
25         }
26     }
27
28     if (Graph[0].cat)
29         DFS_cats(0, Graph, visited, 1, allowed_consecutive_cats);
30     else
31         DFS_cats(0, Graph, visited, 0, allowed_consecutive_cats);
32
33     int count = 0;
34     for (size_t i = 1; i < Graph.size(); i++)
35     {
36         if (Graph[i].connections.size() == 1 && visited[i])
37             count++;
38     }

```

A lap eredeti címe: „https://algowiki.miraheze.org/w/index.php?title=Codeforces_580C_Kefa_and_Park&oldid=1303”