# Feladat

Adott egy ***n*** csúcsból álló irányítatlan fa, ahol „**A”** és **„B”** addig lépkednek a fa csúcsain felváltva, amíg „**A”** ugyanarra a csúcsra nem ér, mint **„B”**. „**A”** a gyökér elemből (1), **„B”** az ***„x”*** csúcsból indul. **„B”** lép elsőnek.

* **2 ≤ *n* ≤ 2·105**
* **2 ≤ *x* ≤ *n***

## **Az eredeti feladat**

<https://codeforces.com/contest/813/problem/C>

# Megoldás

## **Fontos gondolatok**

Legyen bármely két szomszédos csúcs távolsága 1 egység. Azt kell számon tartanunk, hogy „**A”** -nak hány lépésbe telik elérni **„B”** -t, majd ezt megszorozni 2-vel, mivel minden körben mindkettőjük lép (ha egyikük nem lép sehová, az egyhelyben lépésnek számít).

A minimum távolság, amit „**A”** -nak meg kell tennie, az a gyökér elem és az **„x”** csúcs távolsága. Ehhez hozzá kell adnunk azt a távolságot, amit ez idő alatt **„B”** megtett.

## **Részletes megoldás**

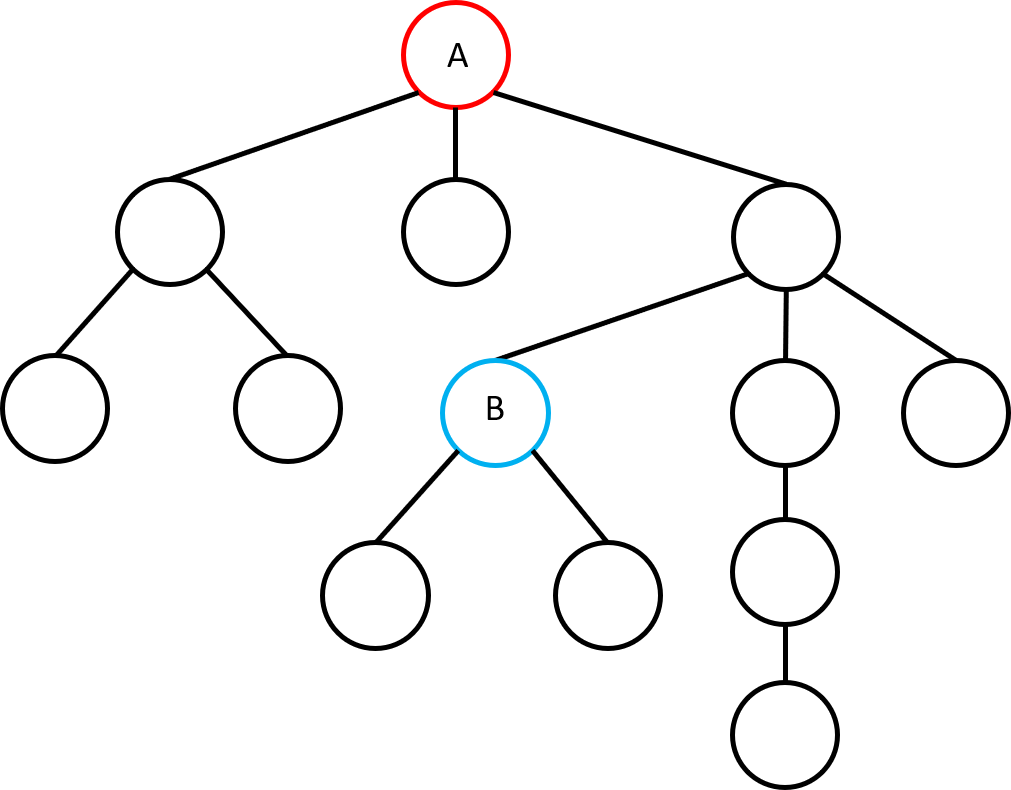
### A célok tisztázása

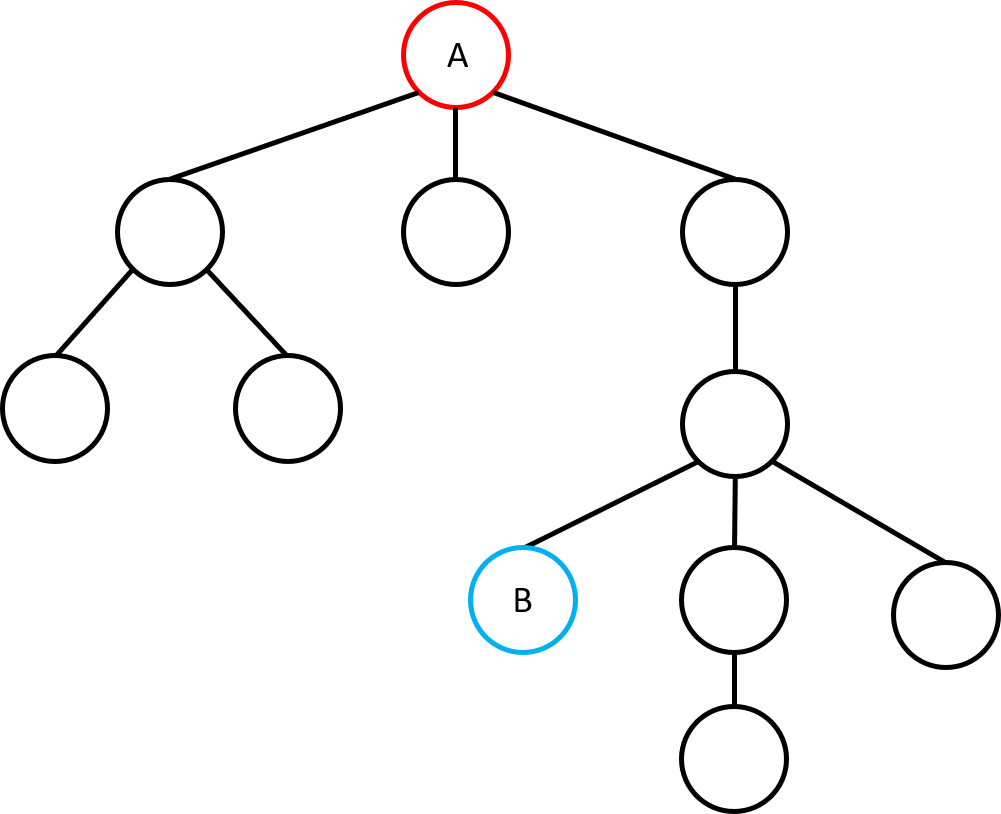
A célja minden alkalommal egyre közelebb jutni **„B”** -hez.

**„B”** célja az „**A”** -tól való távolság maximalizálása, nem az hogy minden lépésben távolodjon „**A”** -tól. Ha az „**A”** -tól való legnagyobb távolság eléréséhez egy felfelé („**A”** felé) történő lépés szükséges, akkor *megfelelő feltételek* teljesülése esetén ezt meg kell tenni.

### A felfele lépés szükséges feltételei

**„B”**  akkor léphet felfelé („**A”** felé) a „d” csúcsra, ha „**A”** a következő lépésével még nem éri el a „d” csúcsot, tehát „**A”** nak a d csúcstól vett távolsága legalább eggyel nagyobb mint **„B”**-nek a „d” csúcstól vett távolsága.





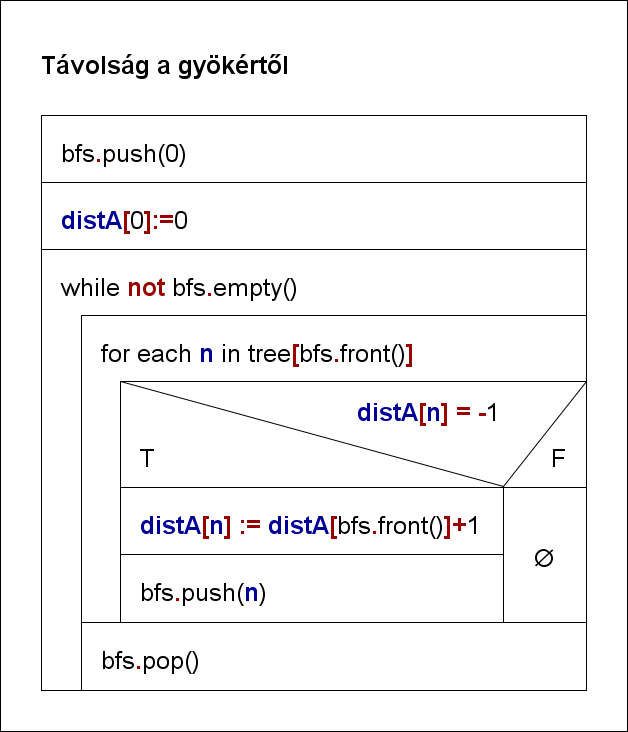
### A megoldás menete

Tároljuk el a fát szomszédsági lista segítségével. Erre mélységi bejárásokat alkalmazva határozzuk meg a lépések számát.

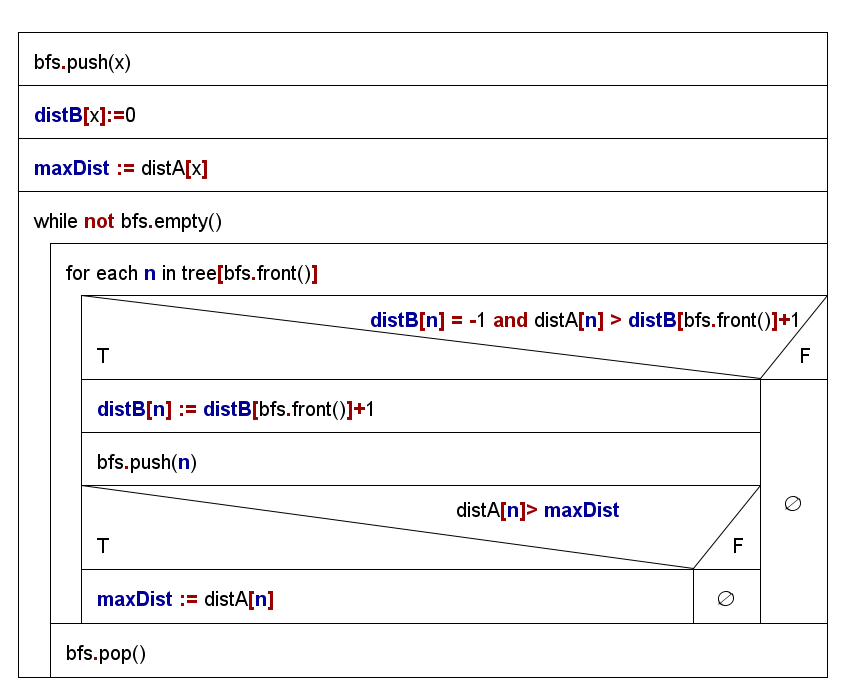
Hogy a **„B”** felfelé lépésére vonatkozó szükséges feltételt vizsgálni tudjuk, először mélységi bejárás segítségével határozzuk meg az összes csúcs „**A”** kiindulási helyzetétől, tehát a gyökér elemtől, való távolságát.

tree: a fa szomszédsági listás reprezentációban

bfs: sor, a mélységi bejáráshoz

distA: tömb, i. eleme a gyökér i.csúcstól vett távolsága, alapból -1 az összes eleme (ez jelzi, hogy még nem vizsgáltuk) 

Tároljuk az ***„x”*** csúcs, és a gyökér elem távolságát a ***maxDist*** változóban. Ez után számoljuk ki az összes olyan csúcs távolságát az ***„x”*** csúcstól, ahová **„B”** léphet, és eközben állítsuk be ***maxDist***-et az előforduló legnagyobb értékre.



A feladat megoldása ***maxDist***\*2.

# Komplexitás

O(2N)

# Implementáció

## C++ implementáció

<https://pastebin.com/ADTSxbTm>