

1. Bemenet és adatstruktúrák:

- A fa csúcsainak értékei egy `k[]` tömbben vannak tárolva.
- Az éleket egy `graph[]` vektorban tároljuk, ahol minden csúcs szomszédait tartalmazza.
- A `fenwickTree[]` tömb az egyes csúcsok al-fájának összegzéseit kezeli. A `update()` és `query()` függvények ezt használják.

2. DFS bejárás:

- Az `dfs()` függvény segít a fa bejárásában, és az `inTime[]` és `outTime[]` tömbökkel követjük, hogy mikor lépünk be és mikor hagyjuk el egy csúcsot a DFS során. Ez az időpont azonosítja az al-fák csúcsait.
- A bejárás idején az `inTime` és `outTime` adatok segítségével jelöljük, hogy mely csúcsok tartoznak egy adott al-fához.

3. Frissítés és lekérdezés:

- A `update()` függvény frissíti a fenwick fát egy adott indexnél, amely az al-fához tartozó csúcsok értékeinek módosítását jelenti.
- A `query()` függvény kiszámítja egy adott index alatti csúcsok értékének összegét.

4. Lekérdezések:

- Ha egy `1 s x` lekérdezés érkezik, akkor a csúcs értéke a fenwick fában frissül.
- Ha egy `2 s` lekérdezés érkezik, akkor az adott csúcs al-fájában lévő összes csúcs értékét kérdezzük le a fenwick fa segítségével. Az al-fát az `inTime[]` és `outTime[]` alapján azonosítjuk.

Lépések a megoldásban:

1. **Bemenet feldolgozása:** Beolvassuk a fa csúcsainak számát (`n`), a lekérdezések számát (`q`), a csúcsok értékeit és az éleket.
2. **DFS bejárás:** A fa bejárásával kiszámítjuk a `inTime[]` és `outTime[]` tömböket.
3. **Frissítés és lekérdezés feldolgozása:** A `1 s x` lekérdezés esetén frissítjük a csúcs értékét a fenwick fában, a `2 s` lekérdezés esetén pedig kiszámítjuk a csúcs al-fájának összegét.