

- **Bemenet beolvasása:**

- A_n és m értékeket beolvassuk, ahol n a szintek száma, m pedig az élek száma.
- Két lista készül:
 - `edge`: az élek, ahol `edge[a]` tartalmazza azokat a szinteket, amelyekre közvetlenül el lehet jutni a -ból.
 - `backedge`: az élek visszafelé, ahol a `backedge[b]` tartalmazza azokat a szinteket, amelyekből közvetlenül el lehet jutni b -be.
- Az `in_degree` lista a beérkező élek számát tárolja minden szinthez.

- **Inicializálás:**

- Az `dp` lista tárolja az 1-es szintről a i -es szintre történő eljutás módjainak számát.

- **Élek beolvasása:**

- A bemenet alapján feltöltjük az `edge` és `backedge` listákat, illetve frissítjük az `in_degree` értékeket, hogy tudjuk, mely szintekhez tartozik beérkező él.

- **Topológiai rendezés:**

- A `q` lista segítségével végrehajtjuk a topológiai rendezést. Az `in_degree` értékek alapján a kezdőszintek (amelyekhez nincs beérkező él) bekerülnek a `q` sorba.
- A sorból folyamatosan eltávolítjuk a szinteket, és azok összes szomszédját (élekkel elérhető szinteket) feldolgozzuk.
- Minden szinthez hozzáadjuk a rá mutató szintek eljutási módjainak számát, azaz:
$$dp[node] = (dp[node] + dp[prev_node]) \% MOD.$$

- **Eredmény kiírása:**

- Az `dp[n]` tartalmazza az összes lehetséges módot, ahogyan az 1-es szintről eljuthatunk az n -es szintre.