Explicación Solución

El problema corresponde a encontrar la cantidad de caminos de largo N en el siguiente grafo (se usa la matriz de adyacencia como representación):

$$\begin{pmatrix} 0 & 2 & 2 & 1 \\ 2 & 0 & 0 & 1 \\ 2 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \tag{1}$$

Para facilitar el cálculo se busca una función recursiva de tres argumentos n, u y v, donde $f(n, u, v)$ representa la cantidad de caminos de largo n que comienzan en u y terminan en v (se puede usar una función que solo considere donde terminan los caminos de largo n), con esto se ve que la respuesta para algún n en particular sería la suma de $f(n, u, v)$ para cada $u, v \in G$. Se nota que $f(u, v, 0) = 1$ si y solo si $u = v$, por lo que tenemos un caso base. Dado eso tambien notamos que dado un camino de largo $n-1$ que comienza en u y termina en v', este se puede extender a un camino de largo n que comienza en u y termina en v si y solo si hay al menos una arista que conecte v' con v, más aún la cantidad de formas de extender el camino es exactamente la cantidad de aristas entre v' y v. Dado lo anterior podemos ver que $f(n, u, v)$ es la suma de los $\#e_{(u',v)} \cdot f(n-1, u, u')$, donde $\#e_{(u',v)}$ es la cantidad de aristas entre u' y v. Explicitamente se tiene la siguiente relación:

$$f(n, u, v) = \sum_{u' \in G} \#e_{u',v} \cdot f(n-1, u, u'). \tag{2}$$

Con lo anterior vemos que hay una primera forma de calcular el resultado final, de forma top-down esto nos da una complejidad $O(n)$ con una tabla $O(n)$ elementos, para el siguiente pasó notemos que $f(n, u, v)$ solo depende de los valores de $f(n-1, u, u')$, en otras palabras si queremos calcular $f(n, \cdot, \cdot)$ solo necesitamos tener precalculado $f(n-1, \cdot, \cdot)$, por lo que usando un approach bottom-up llegamos a una complejidad $O(n)$ con una tabla de $O(1)$ elementos. Para el último paso es necesario darse cuenta de que la ecuación (2) es lineal, más aún podemos reexpresar f en terminos de una función g donde $g(n)$ es una matriz donde $f(n, u, v) = g(n)_{uv}$, usando esto llegamos a la expresión

$$g(n+1) = A \cdot g(n), \tag{3}$$

donde A es la matriz de (1)[1]. Aquí se nota que, por inducción, se tiene que $g(n) = A^n \cdot g(0)$, y recordando la definición de $f(0, \cdot, \cdot)$, vemos que $g(n) = A^n$. Ahora, se puede usar el algoritmo de exponenciación binaria para calcular $g(n)$ con una complejidad $O \log(n)$ y usando $O(1)$ espacio extra.

[1]Esto se puede verificar haciendo lo cálculos explicitos.