12. $\star$Demostrar que las representaciones por listas de adyacencias de un grafo (ejercicio anterior) se pueden construir en $O(n+m)$ tiempo. ¿Qué ocurre si se usa una tabla de hash? ¿Y si se construye una matriz de adyacencias?

(item anterior)

N se representa con una secuencia (vector o lista) que en cada posición v tiene el conjunto $N(v)$ implementado sobre una secuencia (lista o vector). Cada vértice es una estructura que tiene un índice para acceder en $O(1)$ a $N(v)$. Esta representación se conoce comúnmente como *lista de adyacencias*.

ídem anterior, pero cada $w \in N(v)$ se almacena junto con un índice a la posición que ocupa v en $N(w)$. Esta representación también se conoce como *lista de adyacencias*, pero tiene información para implementar operaciones dinámicas.

Demostración

Sea $G = (V, E)$ con $|V| = n, |E| = m$

Y sabemos que $\sum_{v \in V} d(v) = 2m$

Inicializar un vector para cada $v \in V$ es $O(|V|) = O(n)$, luego agregar cada vecino de v es $O(d(v))$, pero a lo sumo hago esta operación $2m$ veces, luego agregar cada vecino de cada $v \in V$ es $O(2m) \in O(m)$, por lo tanto, es $O(n + m)$