Este ejercicio fue hecho por G16 (Luis Pereira y Eric Ruiz Diaz) y con la ayuda y cooperación de G03 (Camila Alderete)

Ejercicio 3

1. Es recomendable usar lista de adyacencia ya que ocuparía un espacio de |V| + E; en este caso 30.000, a diferencia de la matriz de adyacencia que ocuparía un espacio de |V|^2, o de 100.000.000;
2. Para este caso, ya que no tenemos un parámetro en específico que cumplir veremos cómo se comparan para cada función, como tenemos 10.000 vértices y

20.000.000 de aristas, consideraremos el grado de cada arista con un promedio de 2.000;

|  |  |  |
| --- | --- | --- |
|  | Matriz de adyacencia | Lista de adyacencia |
| Espacio | 100.000.000 | 20.010.000 |
| Aristas Incidentes | 10.000 | 2000 |
| Son Adyacentes | 1 | 2000 |
| Insertar Vértice | 100.000.000 | 1 |
| Insertar Arista | 1 | 1 |
| Remover Vértice | 100.000.000 | 2000 |
| Remover Arista | 1 | 4000 |

Según estos datos puedo concluir que la operación más costosa de la lista de adyacencia seria Remover arista con un rendimiento de 4000, y el espacio seria de 20.010.000; a diferencia de la matriz que tiene como operación más costosa remover vértice e insertar vértice, con un rendimiento de 100.000.000 cada uno y un costo espacial de 100.000.000.

Se puede deducir con estos resultados que la lista de adyacencia es más conveniente, pero para asegurar veremos si el grafo es denso o es esparcido.

Un grafo completo por definición tiene n\*(n-1)/2 aristas, donde n es el número de vértices. Remplazando los datos tendríamos que (10.000\*(9999)/2) = 49.995.000, El grafo tiene menos de la mitad de aristas necesarias para considerarse un grafo denso, y si llegásemos a utilizar una matriz de adyacencia más de 50% del espacio utilizado seria desperdiciado, Ahora sí se puede concluir que para este caso una Lista de Adyacencia es la representación más conveniente.

1. En este caso convendría una matriz, ya que para preguntar si una arista (i, j) existe en un grafo solo necesitamos ir a la posición (i, j) de la matriz y ver si existe (1); caso contrario con la lista de adyacencia necesitaríamos ir a todos los nodos adyacentes al nodo i, y ver si uno de ellos es j, cuyo peor caso sería grado (i).
2. Para este caso convendría usar una lista de adyacencia ya que el tiempo seria (1), ir a ese nodo y ver si tiene otro nodo al cual está conectado, a diferencia de la matriz de adyacencia que tendría un peor tiempo de |V| si es que ese nodo está conectado con el ultimo en la matriz.