حل اسأله التكليف الثاني:

السؤال: وضحي بالتحليل طرق تمثيل الـ Graphs برمجياً، مع المقارنة بين الـ Adjacency Matrix والـ Adjacency List:

لتمثيل الرسوم البيانية (Graphs) داخل الحاسب، يوجد أسلوبان أساسيان يختلفان في الكفاءة وطريقة التخزين:

١. مصفوفة الجوار (Adjacency Matrix)

- التعريف: هي عبارة عن مصفوفة مربعة (٢D Array) أبعادها تعتمد على عدد العقد (Vertices). يتم تمثيل الارتباط بين أي عقدتين بوضع قيمة (مثل ١ أو ٠) في الخانة المتقاطعة بين الصف والعمود.
- المميزات: تسمح بالوصول المباشر والسريع جداً للتحقق من وجود حافة (Edge) بين أي عقدتين، حيث تستغرق العملية وقتاً ثابتاً O(١).
- العيوب: تستهلك مساحة تخزينية كبيرة من الذاكرة (Memory) لأنها تحجز مكاناً لكل الاحتمالات، مما يجعلها غير فعالة في حالة الـ Sparse Graphs (الرسوم ذات الروابط القليلة).

٢. قائمة الجوار (Adjacency List)

- التعريف: تعتمد هذه الطريقة على إنشاء مصفوفة من القوائم المرتبطة (Linked Lists). كل عقدة في الرسم البياني تمتلك قائمة خاصة بها تحتوي فقط على العقد المجاورة (Neighbors) المرتبطة بها فعلياً.
- المميزات: توفر في مساحة الذاكرة بشكل كبير، حيث لا يتم تخزين سوى الروابط (Edges) الموجودة حقيقةً، وهي مثالية للتعامل مع الـ Sparse Graphs.
- العيوب: البحث عن وجود اتصال محدد بين نقطتين قد يكون أبطأ مقارنة بالمصفوفة، لأننا قد نحتاج للمرور عبر عناصر الـ Linked List للعثور على الهدف.