



الفرق بين Singly Linked List ، Circular Linked List ، Doubly Linked List

مناسبة لكل من هذه ، استخدام كل منها ، عيوب كل منها :-

① Singly Linked List : بسيطة وسهلة التنفيذ ② تتصلك ذاكرة أفضل ③ إخراج عنصر
سريع بحدوث الحاصلة بزيادة عناصر ④ مناسبة للتطبيقات البسيطة .
عيوبها : لا يمكن الوصول للعنصر ⑤ البحث فيها بطيء ⑥ يجب
الاحتياط في الذاكرة ⑦ عدم الوصول العنصري
الاستخدامات : ① Queue ② Stack ③ عند الحاجة العودة للعنصر

④ linked and hash : احتياط ، للاطلاع والعنصر ⑤ الحذف أسرع ⑥ مناسبة للتتبع
ثنائي الاتجاه

عيوبها : تتصلك ذاكرة أكبر ② أخذ في التنفيذ ③ تصاعدي ترتيب
الاضافة والحذف

الاستخدامات : ① Undo/Redo ② مخرجات الويب ③ أنظمة الملاحة

④ circular linked : لا يوجد فيها null ⑤ مناسبة للتكرار المستمر ⑥ يمكن
البدء من أي نقطة ⑦ فعالة في التطبيقات الدورية
عيوبها : قد يشب دمجها ⑧ أصعب في التتبع ⑨ تحتاج ترتيب وضع
الاستخدامات : ⑩ قوائم تشغيل مشغول ⑪ ألعاب التتبع الدور ⑫ إدارة
الزوار السويك .



أذكر الآن بين Adjacency matrix and Adjacency list

Adjacency list : فكلها لكل vertex نخزن قائمة بالذي يتصل به
هناكها

نستخدم ذاكرة أقل @ مساحة التخزين (إنها غير متواضعة) @ فعالة في خوارزميات مثل DFS & BFS @ مناسبة للرسوم الكبيرة
عندما:

@ فحص وجود Edge بين رأسين أبدا @ المتغير الحقد من المصفوفة
@ تصاريح عميل كل إمكانيات (list)

تتفهمها @ التكرار @ خوارزميات الطرق @ الرسوم فكلها الصفوف

Adjacency matrix : فكرتها مصفوفة ثنائية الأبعاد $V \times V$ ، القيمة 1 تعني
وجود حافة لها.

هناكها:

@ فحص وجود الحافة سريع جدا @ تنفيذ بسيط وسهل @ مناسب للرسوم الكبيرة
عندما:

@ تتكامل ذاكرة أكبر @ غير فعالة للرسوم فكلها الحواف @ تضاعف المساحة المستخدمة
تتفهمها @ الرسوم المعقدة @ مسائل تتطلب فحص جميع الحواف