
Theoretical Questions

What are the differences between Singly Linked List, Circular Linked List, and Doubly Linked List?

(1) Singly Linked List (القائمة المرتبطة الأحادية)

Uses (الاستخدامات):

تُستخدم عندما نحتاج تخزين بيانات بشكل ديناميكي (حجم متغير).
مناسبة للتنقل في اتجاه واحد فقط (من البداية للنهاية).

تُستخدم في تنفيذ:

Stacks

Queues

Undo operations (بشكل بسيط)

Pros (المميزات):

تستهلك مساحة أقل من Doubly Linked List.

سهولة التنفيذ والفهم.

لا تحتاج إلى مساحة إضافية لمؤشر سابق.

Cons (العيوب):

لا يمكن التنقل للخلف (اتجاه واحد فقط).

حذف أو إدراج عقدة يتطلب الوصول للعقدة السابقة.

البحث أبطأ مقارنة بالمصفوفات ($O(n)$).

(2) Circular Linked List (القائمة المرتبطة الدائرية)

Uses (الاستخدامات):

تُستخدم عندما نحتاج تكرار المرور على العناصر بدون نهاية.

مناسبة في:

Round Robin Scheduling

Multimedia playlists

الألعاب (الدوران بين اللاعبين)

Pros (المميزات):

لا يوجد عنصر أخير يشير إلى null.

يمكن البدء من أي عقدة والتنقل لجميع العناصر.

مفيدة للتطبيقات التي تحتاج تكرار مستمر.

Cons (العيوب):

أصعب في الفهم والتنفيذ مقارنة ب Singly Linked List.

قد تؤدي إلى حلقات لا نهائية إذا لم يُعامل معها بحذر.

عمليات الحذف والإدراج تحتاج انتباه إضافي.

3) Doubly Linked List (القائمة المرتبطة الثنائية)

Uses (الاستخدامات):

تُستخدم عندما نحتاج التنقل للأمام والخلف.

مناسبة في:

Browser history (Forward / Backward)

Undo & Redo operations

Music players

Pros (المميزات):

يمكن التنقل في الاتجاهين (forward & backward).

تسهّل عمليات الحذف والإدراج.

لا تحتاج معرفة العقدة السابقة عند الحذف.

Cons (العيوب):

تستهلك مساحة أكبر بسبب وجود مؤشرين (prev & next).

أكثر تعقيدًا في التنفيذ.

تحتاج وقتًا أطول قليلًا في التحديث.

Summary (ملخص سريع)

النوع	استهلاك. الذاكرة	سهولة التنفيذ
Singly.	اتجاه واحد قليل	سهل
Circular	دائري متوسط متوسط	
Doubly	اتجاهين عالي. أصعب	
