

حل التكليف الأول: هياكل البيانات (Data Structures)

المقارنة بين أنواع القوائم المرتبطة (Linked Lists):

تُعد القوائم المرتبطة من أهم هياكل البيانات الديناميكية، وتنقسم إلى عدة أنواع لكل منها خصائص محددة:

١. القائمة المرتبطة البسيطة (Singly Linked List)

الوصف: تتكون من سلسلة من العقد (Nodes)، حيث تحتوي كل عقدة على البيانات ومؤشر واحد (Next) يوجهنا للعقدة التي تليها فقط.

الاستخدامات:

بناء هياكل البيانات الخطية مثل المكدس (Stack).

إدارة الذاكرة في الأنظمة البسيطة التي تتطلب تصفحاً في اتجاه واحد.

المميزات:

كافأة عالية في استهلاك الذاكرة (تخزن مؤشراً واحداً لكل عقدة).

سرعة تنفيذ عمليات الإضافة والحذف في بداية القائمة.

العيوب:

محدودية الحركة؛ حيث لا يمكن الرجوع للعقدة السابقة.

صعوبة الوصول لعقدة معينة دون المرور بجميع العقد السابقة لها.

٢. القائمة المرتبطة المزدوجة (Doubly Linked List)

الوصف: تتميز بأن كل عقدة تحتوي على مؤشرين: أحدهما للعقدة التالية (Next) والأخر للعقدة السابقة (Previous).

الاستخدامات:

أنظمة سجل الزوار في المتصفحات (للرجوع للخلف وللأمام).

خاصية "التراجع" (Undo) و "الإعادة" (Redo) في التطبيقات.

المميزات:

مرنة كاملة في التنقل (تصفح ثانوي الاتجاه).

سهولة حذف عقدة معينة لتوفر مرجع مباشر للعقدة التي تسبقها.

العيوب:

تحتاج مساحة أكبر في الذاكرة لتخزين المؤشر الإضافي.

زيادة التعقيد البرمجي عند إجراء عمليات الإضافة أو الحذف.

٣. القائمة المرتبطة الدائرية (Circular Linked List)

الوصف:

هي نوع من القوائم (سواء كانت بسيطة أو مزدوجة) حيث تشير العقدة الأخيرة فيها إلى العقدة الأولى، مما يوجد حلقة مستمرة.

الاستخدامات:

جدولة المهام في أنظمة التشغيل (خوارزمية Round Robin).
تطبيقات الألعاب التي تتطلب تبادل الأدوار بشكل دوري.

المميزات:

إمكانية الوصول لأي عقدة انتلاقاً من أي نقطة في القائمة.

مثالية للتطبيقات التي تعمل بشكل دوري مستمر.

العيوب:

تتطلب حذراً برمجياً لتجنب الدخول في حلقات تكرار لا نهائية (Infinite Loop).
صعوبة تحديد نهاية القائمة بشكل دقيق دون منطق برمجي إضافي.

