



データ構造とアルゴリズムを使用した不正検出のための金融取引の分析

(実施期間: 5週目の月曜日から10週目の月曜日 - 30 マーク)

背景

最近、中堅デジタル銀行の複数の顧客口座からの不正取引に絡む重大な金融詐欺事件が明るみに出た。この不正行為は数時間にわたって発覚せず、その間に200万ドル以上が吸い上げられた。この事件は規制当局の監視を促し、オンライン・バンキング・プラットフォームのセキュリティ・インフラの堅牢性についての懸念を高め、より高度な不正検知システムと顧客意識の向上が急務であることを強調した。

金融機関における重要な課題のひとつは、**膨大な金融取引データを効率的に管理・処理すること**です。**特定の取引を検索し、大規模なデータセットを効率的に並べ替える能力は**、業務運営の最適化にとって極めて重要です。

データ構造はこれらのプロセスにおいて重要な役割を果たす。**配列とリンクリストは**、財務データの保存と管理に使用できる基本的なデータ構造であり、異なる**検索とソートのアルゴリズムは**、情報の検索と分析の効率を決定する。

この課題では、金融取引データを整理する際に、さまざまな検索とソートのアルゴリズムがどのように実行されるかを評価し、配列やリンクリストなどのデータ構造の選択が、時間とメモリ効率の面でパフォーマンスにどのように影響するかを探ることを目標とする。

質問だ:

この課題では、**データセット**が提供されます:

- **financial_fraud_detection.csv** - 不正検知研究のために実世界の行動をシミュレートするために生成された金融取引が含まれる。(トランザクションの詳細、行動の特徴およびメタデータ、詐欺インジケータ)。

あなたは、金融取引を格納するために、配列ベースとリンクベースを使用してデータ収集クラスを開発する必要があります。その後、適切な検索とソートのアルゴリズムを実装し、以下の機能要件を満たす必要があります。

1. 金融取引を決済チャネルごとに分けて保存
2. 金融取引を場所ごとに昇順に並べ替える
(注: データ構造に応じて効率的なソートアルゴリズムを選択すること)
3. 特定の取引タイプを検索する
(注: データ構造に応じて効率的な検索アルゴリズムを選択すること)
4. 特定のデータコレクションからJSONフォーマットを生成する
(ヒント: C++を使ってデータをJSONにエクスポートするには、*nlohmann/json*のようなサードパーティ・ライブラリが使えます)
5. 上記1)~4)の機能を駆動するメインプログラム

* 機能要件に関する論理的な仮定は、どのようなものでも受け入れることができる。

ラボワークの最低条件1

1. 各グループのメンバーは4名までとする。
2. この課題では、2つのうちのどちらか、あるいは両方を開発する必要がある：
 - a) **配列の実装**
 - a) 金融取引を保存し処理するために、**配列ベースのデータ構造**を実装する。
 - b) トランザクションのソート、検索、フィルタリングなどの操作を実行する。
 - b) **リンクされたリストの実装**
 - a) 同じデータを格納する**リンクリストベースのデータ構造**を実装する。
 - b) 配列ベースのアプローチと同様に、ソートや検索などの効率的な操作を可能にする。
3. **選別の条件:**
 - データ構造に適した**ソートアルゴリズム**を使用して、金融取引を**場所別**にソートする。
4. **検索条件**
 - **検索アルゴリズム**を実装し、特定の条件に基づいて関連するトランザクションをフィルタリングして検索する。
5. **パフォーマンス比較:**
 - に対するサーチとソートのアルゴリズムの時間とメモリ効率を比較する。
配列とリンクされたリスト
6. **エラー処理:**
 - 予期しないフォーマットやデータの欠落に対処するために、**堅牢なエラー処理とデータ検証**を実装する。
 - 。

提出要項 #1: プログラムとビデオ提出 - ラボワーク #1 (15点)

1. **プログラミング言語:** C++を使用して両方のプログラムを開発する。
2. `<vector>`や`<list>`のような**組み込みコンテナはなく**、自分でデータ構造を作る必要がある。
3. **ZIP送信:**
 - .cpp、.hpp、.csv/textファイルのみを含めます。
 - このファイル名の形式に従ってください:
`<グループNo>_<チームリーダーID>_<第1メンバーID>_<第2メンバーID>_<第3メンバーID>.zip`
 - 例えば、`"G1_TP012345_TP014556_TP067554_TP034325.zip"`
4. **ビデオ録画:**
 - また、各チームリーダーは録画ビデオを**1本**アップロードしなければならない。
 - 全員参加（1人5分以内）。
 - 各メンバーは、ワード文書に記載されている仕事量マトリックス配分表と関連付けて説明しなければならない。
 - 最終的に録画したビデオを**200MB以下**に圧縮してから提出すること。
 - 指定された制限時間（30分）を超える場合は、指定された時間までしか評価されません。
 - 動画は通常速度（1倍速）で録画する必要があり、デモ動画の時間要件を満たすために速度を上げたり調整したりすることはできません。
 - ビデオ録画ファイルは、以下のファイル名形式に従う必要があります:
`<グループNo>_<チームリーダーID>_<第1メンバーID>_<第2メンバーID>_<第3メンバーID>.mp4`
 - 例えば、`"G1_TP012345_TP014556_TP067554_TP034325.mp4"`
5. ラボ評価課題#1の採点基準については、**5 ページ**を参照のこと。

提出要項その2: ドキュメンテーション提出 - ソリューション・ワーク (15点)

文書に記載する内容

1. 表紙**2. 署名入り作業負荷マトリックス表**

この表は、ラボ評価ワーク#1における各メンバーの個人的な最終評点に影響することに留意すること。

3. 理論的説明

- 選択したデータ構造におけるデータ操作を説明する。

4. インプット・アウトプットのスクリーンショット

- システムの入力と出力のスクリーンショットを含む。

5. 総括討議

- システム効率の分析（実行時間、時間効率、スペース効率など）
- 開発中に観察したことをまとめ、議論する。
- コードの長所と短所を批判的に評価する。

6. 結論と反省

- 主な調査結果をまとめる。
- 潜在的な改善点とシステムの弱点について話し合う。
- 課題についての個人的な考えを共有する。

7. 参考文献（該当する場合）

- 外部資料を適切に引用する（APA形式）。

コードを参照しなかった場合、盗作として扱われる。

8. 付録（該当する場合）**提出とフォーマット:**

- ファイル名の命名形式:
<グループNo>_<チームリーダーID>_<メンバー1ID>_<メンバー2ID>_<メンバー3ID>.docx
- 例: G1_TP012345_TP014556_TP067554_TP034325.docx

締め切り:	ムードル
最大ページ数	30ページ
最大公約数:	4500字

採点基準

(ラボ評価ワーク#1 - 15 MARKS)

このラボ評価ワーク#1は、以下のパフォーマンス基準に従って評価される：

評価項目	包括的	15マーク
CLO3：ラボ評価タスク#1 - 30分間のビデオ録画（評価は個人のパフォーマンスに基づいて行われる。）		
実践的なスキルデータ構造とアルゴリズムの使用+ 個人的理解		
データ構造の活用	技術能力	
関連アルゴリズムの実装	技術能力	
使用されるデータ構造/アルゴリズムについて理解していることを示す	包括的な理解	
構造／アルゴリズムの選択を正当化する	洞察に満ちた正当化	

採点基準

(ソリューションワーク - 15 MARKS)

このソリューションは、以下のパフォーマンス基準に従って評価される：

評価項目	包括的	15マーク
CLO2：ソリューションワークドキュメンテーション（評価はグループコンポーネントに基づいて行われる）		
理論的説明 (例：データ構造、アルゴリズム)	明確な説明	
入出力スクリーンショット	適切なスクリーンショット	
総括討議 (包括的な時間と空間の複雑性)	明確で洞察に満ちた分析	
結論と反省 (その他の関連／重要な議論)	明確なハイライトと洞察力	
コンテンツ組織	よく構成された論理的な流れ	

ドキュメントの総ページ数の目安：30ページ（最大）。

ドキュメンテーションのおおよその文字数：4500ワード（最大）