データベースシステム

第1回
ガイダンス
コストの考え方

本日の内容

- データベースシステムで勉強すること
- 成績の方針

まずはアンケートとお知らせ

- 昨年度欅先生の「データベース設計論」を受けなかった人いますか？
  - 基本的にはリレーショナルデータベースとは何かを知っていることを前提とします

データベースシステムで扱う内容

- 基本的には「リレーショナルデータベース」を対象とします
- 「データベース設計論」でやった内容
  - アプリケーションでデータベースを利用するために一通り必要な技術
    - データモデル、問合せ言語、正規化
- それだけでは十分ではないケース
  - 大量のデータを扱う場合
  - たくさんの人がアクセスする場合

授業で扱う内容（その１）

- データベースシステムにおける問合せ処理の仕組み
  - どんな時に必要な知識か？

    大量のデータを高速に検索したい時に必要

  - 例を示してみます
    - 10万行の名簿データで60歳以上の人を探す

      ```
      select * from user where age > 60;
      ```

    - 工夫していないデータベース
    - 工夫を施したデータベース

検索を高速にするには？

- 復習：線形探索と二分探索
  - 以下の配列から9を検索するには？

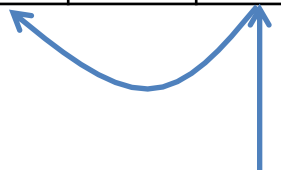
3	10	4	64	34	76	12	9	61	37

- 線形探索の場合…計算コストは n

3	10	4	64	34	76	12	9	61	37

- 二分探索の場合…計算コストは $\log_2 n$

3	4	9	10	12	34	37	61	64	76



データ構造の用意

検索アルゴリズムを適用

検索を高速にするには？

- リレーショナルデータベースの問合せ言語は「非手続的言語」
  - 利用者は何がほしいかを指定する
  - データベースシステムが適切な処理内容をプランニングしてくれる
- しかし、利用者がやる必要のあることがある
  - 使われるだろう検索アルゴリズムを想定して、データ構造を用意しておく必要がある
  - データベースが適切なプランニングをするように利用者が仕向けるのも大事なこと

授業で扱う内容（その１）

- 索引構造と検索アルゴリズム（選択演算）
  - ヒープファイル、ハッシュファイル
  - B+-木
- 結合演算のアルゴリズム
  - ソートマージ結合、ハッシュ結合
  - 外部ソートアルゴリズム
- 問合せ処理のプランニング
  - 問合せのコスト計算見積もり

授業で扱う内容（その2）

- データベースにデータを格納すると…

格納されたデータは責任を持って管理してくれる
（破損したり、無くなったりしない）

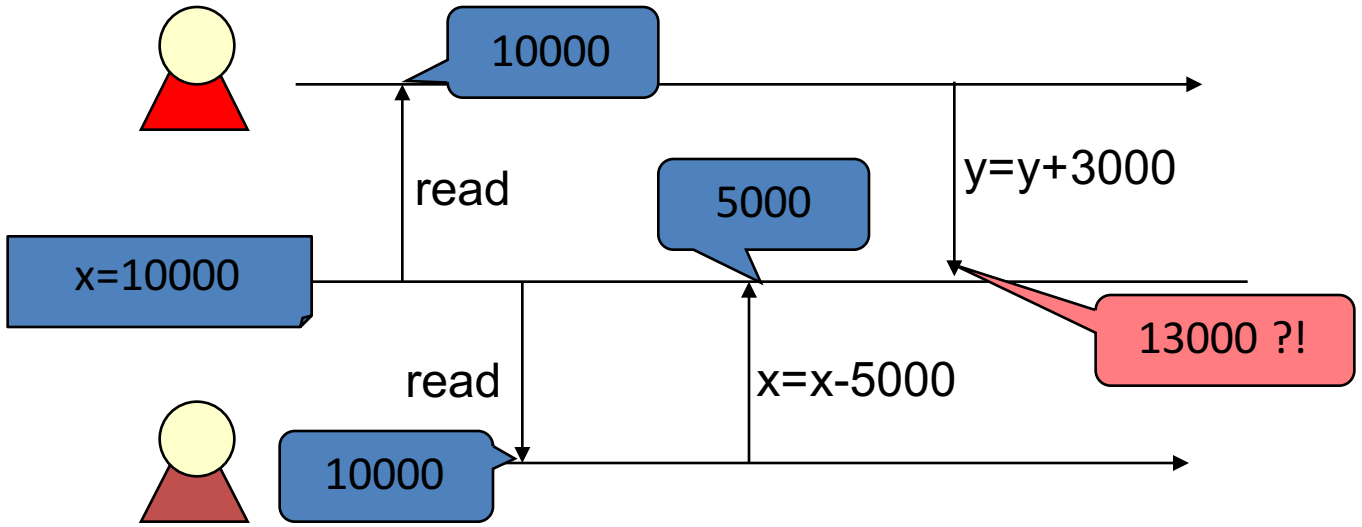
- データを扱う時にはいろんな障害が発生する
  - プログラムが途中でエラーになる
  - データベースが途中でこける
  - システムが故障する
  - ハードディスクが故障する
  - などなど…

授業で扱う内容（その2）

- 様々な障害に対応するためのデータベースの仕組み
  - トランザクション
    - やりかけの処理は破棄し、終わった処理は反映を保障する
  - ログの活用
    - 保存する前に必ずログを取る
    - 回復時にログを使って復旧させるには？

授業で扱う内容（その3）

- 同時実行制御



「トランザクション」（データアクセスの原始的な単位）という概念を導入して管理する

↑トランザクションは障害時回復の際にも利用する

授業で扱う内容（その3）

- ＯＳでも「排他制御」機能がある
  - しかし排他制御はデータベースにとっては厳しすぎる制約
    - 例）終電検索データベースをある利用者が使っているときにほかの人がそのデータベースにアクセスできなくなったら？
  - DBMS専用の同時実行制御
    - 読みと書きでロックの種類を変える
    - プロセスをできるだけ同時に動かしながら、データの一貫性が失われないように制御するには？

授業に関する連絡方法

- slackを使います(https://dbms18.slack.com)
  - 私がMLへの送信権限を持っていないため
  - 連絡や質問もslackでお願いします
    - 質問は個人宛でなく、みんなが見えるように
      （同じ返事を何度も繰り返すのを避けるため）
  - slackの通知機能はONにしておいてください

Slack dbms18 ワークスペースの招待URL

http://goo.gl/JG98tm

授業の資料について

- 授業の資料はGithubにアップします
  - https://github.com/chiemi627/dbms18
  - 前日までにはアップします
  - 当日の授業では資料配布しません
    - 印刷するか、電子的に持ってきてください
    - メモをすることが多いので、書き込めるように用意しておくのが良いでしょう
  - 前の授業の知識をもとに進めることも多いです
    - これまでの授業の資料も見れるようにすると良いです

授業日程について

- GitHubにてお互い確認しましょう

今年度だけの特殊な事情

- 体調により、遠隔講義が開催される可能性があります。
  - 1,2名のボランティアを募ります
    - appear.in を使ってテレビ会議システムで講義
    - PCをプロジェクタに接続、音声出力をスピーカーに接続
    - 質問はslackでリアルタイムに受付
- 急な病欠が起こる可能性もあります
  - 休講のお知らせはslack+工藤先生+非常勤室
  - なんの連絡もなかったらslackで連絡ください

成績について

- 出席点
  - 授業内の演習
- レポート
  - 1，2回ほど出します
- 期末試験

コストの考え方

ちょっとしたゲームをします

内容

1. 高速な検索を実現するために必要な技術
2. 検索時間を考えるための重要な用語
   「IOコスト」
3. もっとも簡単なデータ格納方式
   「ヒープファイル」
4. 問合せの種類別，IOコストの計算の仕方
   ~ ヒープファイルを使って ~
5. IOコストの比較
   ~ ハッシュファイルと比較してみよう ~

問合せの例

- g0720501が履修する授業名を求めるには？

```
SELECT 授業.授業名
FROM 授業, 履修
WHERE 授業.授業番号 = 履修.授業番号
    and 履修.学籍番号 = 'g0720501'
```

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論
AI003	人工知能論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
IT002	g0720502	A
IT002	g0720507	A
AI003	g0720503	B

検索を速くするための技術を学ぶ

- g0720501が履修する授業名を求めるには？

```
SELECT 授業.授業名
FROM 授業, 履修
WHERE 授業.授業番号 = 履修.授業番号
    and 履修.学籍番号 = 'g0720501'
```

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論
AI003	人工知能論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
IT002	g0720502	A
IT002	g0720507	A
AI003	g0720503	B

検索の重要ポイント(その1)

- 条件による選択

```
SELECT 授業.授業名
FROM 授業, 履修
WHERE 授業.授業番号 = 履修.授業番号
    and 履修.学籍番号 = 'g0720501'
```

選択の条件

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
		A
		A
		B

履修テーブルにタプルが1000万あったとき
どうしたら一瞬で検索することができるだろうか？

検索の重要ポイント（その１）

- 条件による選択を高速に行うには
  1. 頻繁に発行される問合せ条件を調べて置くこと
     - アプリケーションで発行される問合せの種類は大体決まっている
  2. 問合せ条件に合った格納方式を選ぶ
  3. 問合せ条件に合った索引を付ける

検索の重要ポイント(その2)

- 結合演算

```
SELECT 授業.授業名
FROM 授業, 履修
WHERE 授業.授業番号 = 履修.授業番号
```

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論
AI003	人工知能論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
IT002	g0720502	A
IT002	g0720507	A

履修テーブルが1000万，授業タプルが1000あっても検索を一瞬で終わらせることができる？

検索の重要なポイント（その２）

- 結合演算
  1. 結合アルゴリズムを知る
  2. 適切な結合アルゴリズムが実行されるように、索引などをあらかじめ用意しておく

補足：授業と履修の結合結果

授業

授業番号	授業名
PR001	プログラミング実習
IT002	情報理論
AI003	人工知能論

履修

授業番号	学籍番号	成績
PR001	g0720501	A
PR001	g0720504	B
IT002	g0720502	A
IT002	g0720507	A
AI003	g0720503	B

授業番号	授業名	授業番号	学籍番号	成績
PR001	プログラミング実習	PR001	g0720501	A
PR001	プログラミング実習	PR001	g0720504	B
IT002	情報理論	IT002	g0720502	A
IT002	情報理論	IT002	g0720507	A
AI003	人工知能論	AI003	g0720503	B

高速な検索を実現するために

データの格納方式と選択処理

- どんな選択条件のときにどの格納方式がよいか

結合処理のアルゴリズム

問合せのプラン組立と最適化

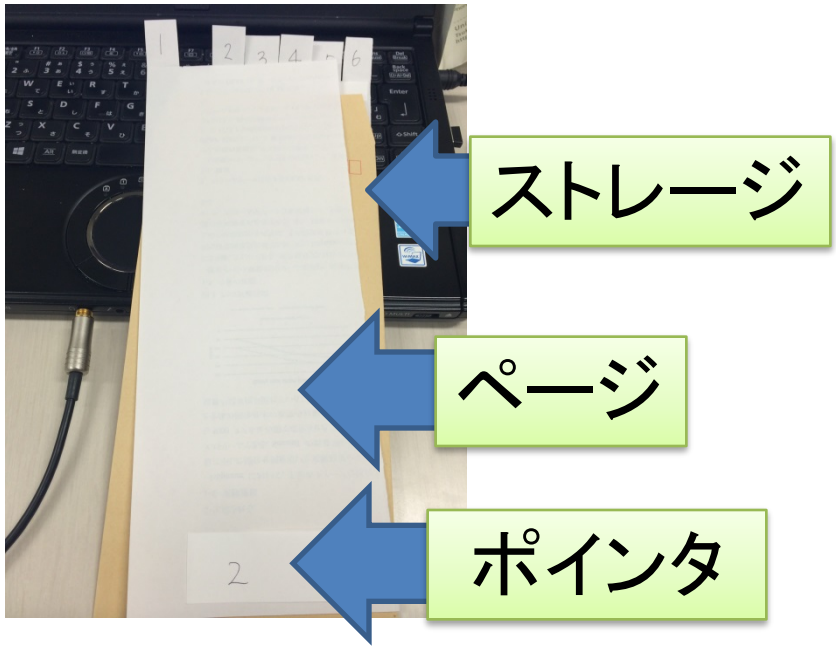
データ格納方式と選択処理

演習のための準備

- 隣＆後ろにいる人で4人組を作りましょう
- 4人組を2人ずつに分けましょう
  - 1組目：データベースを作る
  - 2組目：データを用意する

演習のための準備（その1）

- 簡易データベースシステムを作ります
  - A4用紙3枚をそれぞれ縦半分に切る
  - 上に切れ目を入れて「タブ」を作ってください
    - 1~5まで区別がつくように
  - 一旦封筒に入れます
    - タブ部分を引っ張って引き出すことができればOK



準備2: データベースにデータを挿入する

- 付箋に「名前」と「1～100の数字」を書きます
  - 名前はひらがなで書いてください
  - 数字はランダムに書きましょう
  - 付箋を「レコード」と呼ぶことにします
- レコード20枚分書きましょう
- 1ページに5枚まで貼り付けてデータベースにしまってください
  - ページ1から詰めて貼り付けてください
  - あくまで数字はランダムな並びになるように

演習1：検索速度を測りましょう

- 問合せ
  - 数字が17から24までのレコード数を求めましょう
- 検索のためのルール
  - 検索作業をするのは「データベースを作った」組
    - データを用意した組は時間を測りましょう
  - 封筒の外に出せるページは1ページのみ
    - 次のページを引き出すときは、外に出ているページをしまってから

演習１：振り返り

- 問合せに該当する件数
- かかった時間
- 何処で時間がかかりましたか？
  - なぜ時間がかかったと思いますか？
- 他に気が付いたことがあれば
  書いてみましょう

演習1-2：振り返り

- どのような工夫をしましたか？
- 問合せに該当する件数
- かかった時間
- 最初と比べて速くなった（はず）理由をあげてみましょう

格納するときの基本事項(レコード)

- その1
  テーブルのタプル単位で保存する。
  その時のタプルを「レコード」と呼ぶ

社員番号	社員名	趣味
0650	鈴木一郎	野球
0650	鈴木一郎	盆栽
0650	鈴木一郎	コイン収集
1508	浜崎アユ	作詞

0650	鈴木一郎	野球

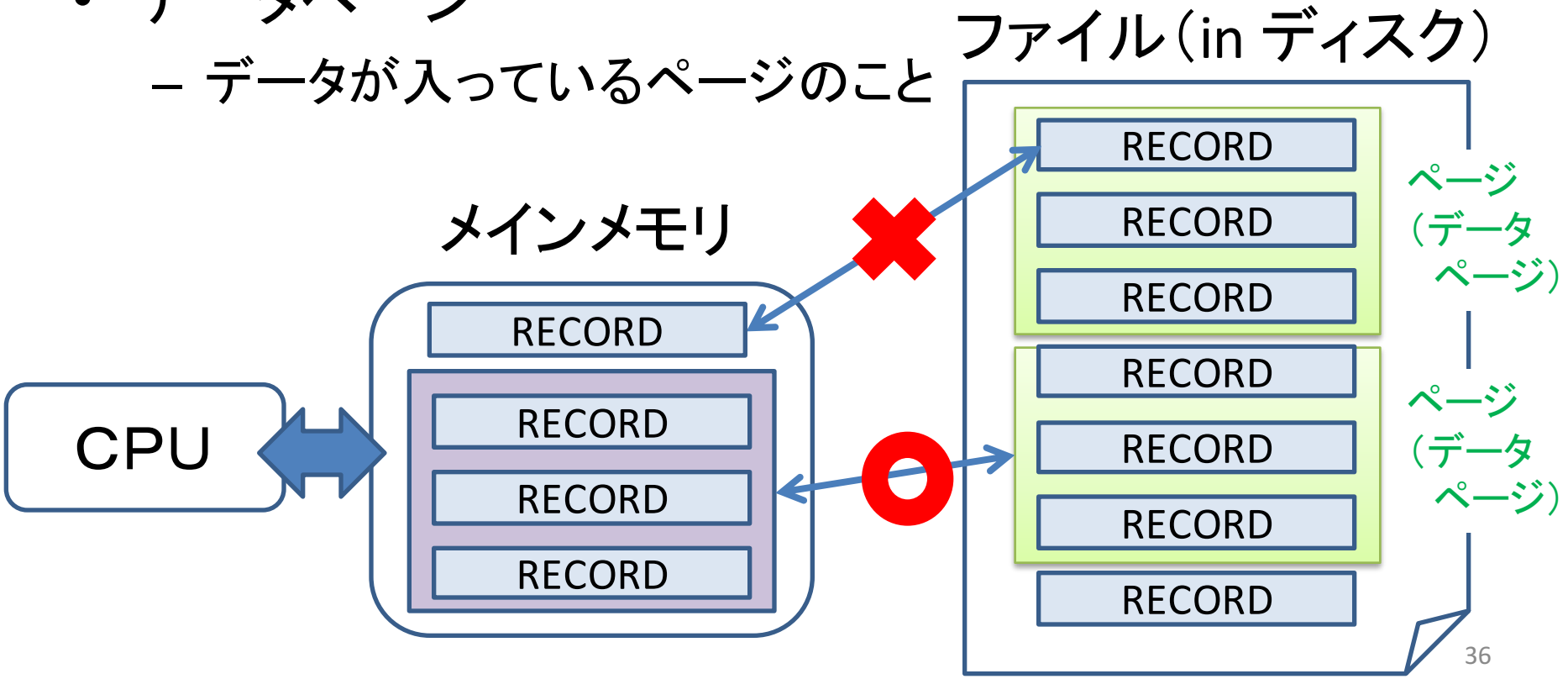
0650	鈴木一郎	盆栽

0650	鈴木一郎	コイン収集

1508	浜崎アユ	作詞

格納するときの基本事項(ページ)

- その2
  ディスク(ファイル)からの読み込みや書き込みはレコード単位では行わず「ページ」単位で行う
- データページ
  - データが入っているページのこと

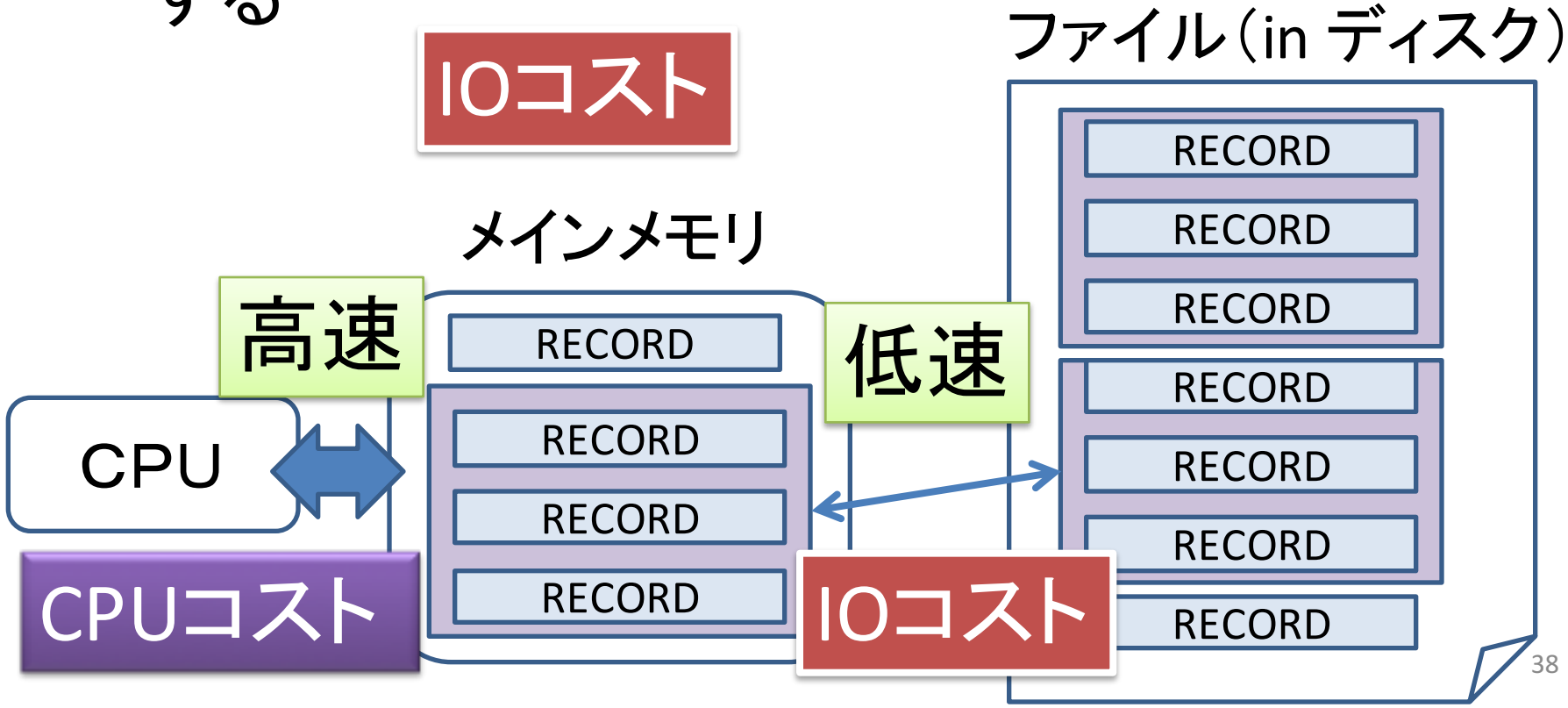


適切なページの大きさ

- デフォルトのページサイズ
  - PostgreSQL 8KB, MySQL 16KB, SQLite 1KB
  - もちろん設定で変更可能
- ページが大きすぎると…
  - たった1つのタプルを読み込みたいだけなのに、たくさんのデータをファイルから読み込まないといけない
- ページが小さすぎると…
  - 全てのタプルを読み込むために何度もディスクにアクセスしなければならない

格納するときの基本事項（IOコスト）

- その3
  アクセスの効率を考えるときはディスク（ファイル）に何回読み書きをしたかを基準にする



演習提出方法