プログラミング応用

http://bit.ly/ouyou3d

アルゴリズム（1）

前期 第13週

2018/7/17

本日は・・・

- データ構造
  - データとは
  - 線形データ構造
- 探索アルゴリズム
  - 線形探索
  - 探索アルゴリズムの高速化

本日は・・・

- データ構造
  - データとは
  - 線形データ構造
- 探索アルゴリズム
  - 線形探索
  - 探索アルゴリズムの高速化

データとは

何かを文字や符号、数値などのまとまりとして表現したもの。

※IT用語辞典より

【例1】ひたちなか市の最高気温

7/17	7/18	7/19	7/20	7/21	7/22	7/23
36	35	34	34	35	34	34

※ウェザーニュースより

【例2】ディレクトリの階層

【例3】Facebookの友達関係

データ構造とは

データの集まりをコンピュータプログラムで処理する際に扱いやすいように、一定の形式で格納したもの

※IT用語辞典より

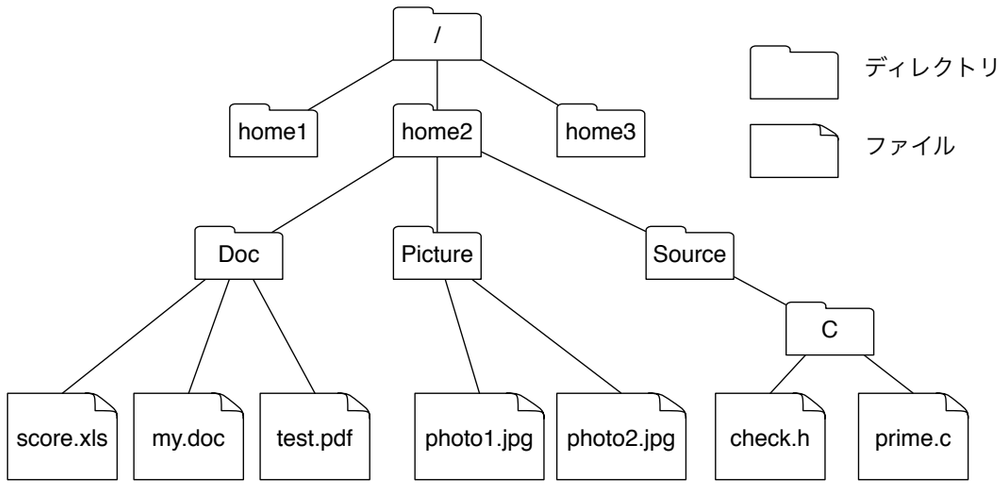
【例1】ひたちなか市の最高気温 → 線形構造

7/17	7/18	7/19	7/20	7/21	7/22	7/23
36	35	34	34	35	34	34

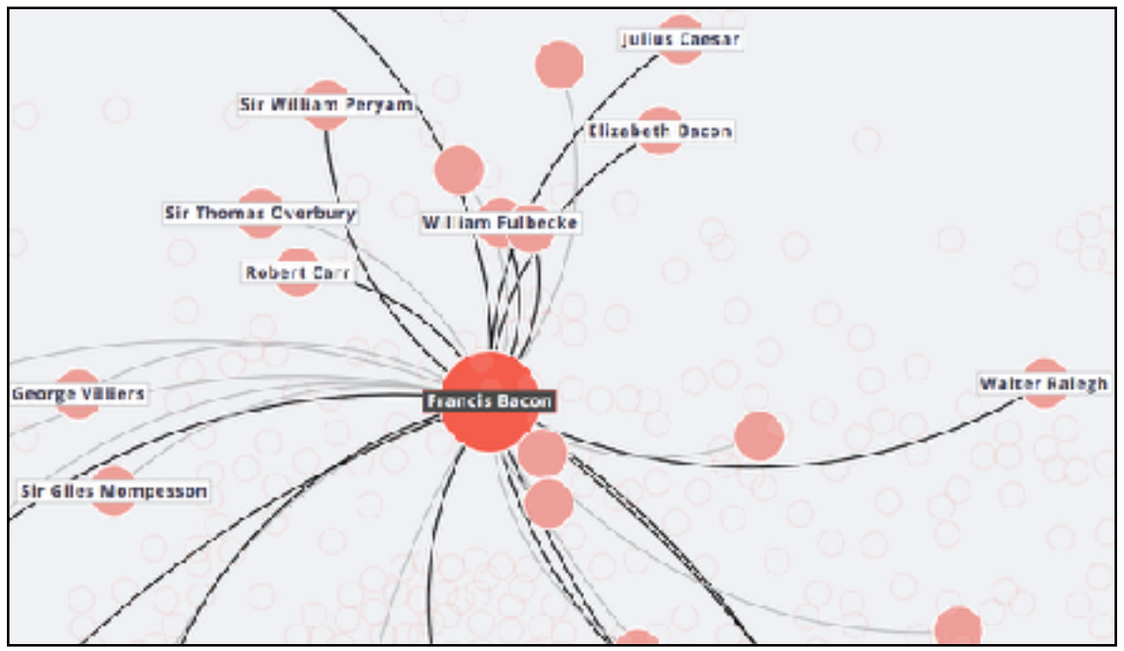
※ウェザーニュースより

データ構造とは

【例2】ディレクトリの階層 → 木構造



【例3】Facebookの友達関係 → グラフ構造



※ http://www.sixdegreesoffrancisbacon.com より

データ構造とは

- **線形構造**　→ **この授業で学ぶ**
- **木構造, グラフ構造など**
  → **データ構造とアルゴリズム（4年生）**

本日は・・・

- データ構造
  - データとは
  - 線形データ構造
- 探索アルゴリズム
  - 線形探索
  - 探索アルゴリズムの高速化

線形構造の表現

配列での表現が一番簡単

今回は、データを区別できるようにラベルを付けてみた

7/17	7/18	7/19	7/20	7/21	7/22	7/23
36	35	34	34	35	34	34

数値のまとまりを配列として格納する

```
//ラベルは文字列の配列として表現
char *dates1[N] = {"7/17", "7/18", "7/19", "7/20", “7/21",
                   "7/22", "7/23"};
//データは整数の配列として表現
int temperature1[N] = {36, 35, 34, 34, 35, 34, 34};
```

線形構造の出力（13_show.c）

文字列（char型のポインタ）の配列を参照渡し

整数の配列を参照渡し

配列のサイズ（要素数）

```
1: #include <stdio.h>
2: #define N 7
3:
4: void show(char *label[], int value[], int size)
5: {
6:     int i;
7:     for(i=0; i<size; i++) {
8:         printf("label: %s, ", label[i]);
9:         printf("value: %d\n", value[i]);
10:    }
11: }
12:
13: int main(void)
14: {
15:   char *dates1[N] = {"7/17", "7/18", "7/19", "7/20", “7/21",
                         "7/22", “7/23"};
16:   int temperature1[N] = {36, 35, 34, 34, 35, 34, 34};
17:
18:   printf("---show---\n");
19:   show(dates1, temperature1, N);
20:
21:   return 0;
22: }
```

本日は・・・

- データ構造
  - データとは
  - 線形データ構造
- 探索アルゴリズム
  - 線形探索
  - 探索アルゴリズムの高速化

探索とは

- 条件を満たすデータを探す処理
  （一致する値を探す、など）
- 探索は多くのソフトウェアで必要になる
  基盤となる処理
  - 検索サービス
    → キーワードを含むページを探索
  - グルメサイト
    → 指定した都道府県飲食店を探索

探索の例

「最初に34度になるのはいつ？」

ひたちなか市の最高気温

7/17	7/18	7/19	7/20	7/21	7/22	7/23
36	35	34	34	35	34	34

この日を出力したい

どんなアルゴリズムで探索する？

線形構造の探索アルゴリズム

- **線形探索**
- **線形探索 ＋ 番兵法**
- **二分探索**
- **などなど・・・**

線形探索

「最初に34度になるのはいつ？」

① この値は35？
② 配列の末尾？ → 次に進む

③ この値は35？
④ 配列の末尾？ → 次に進む

7/17	7/18	7/19	7/20	7/21	7/22	7/23
36	35	34	34	35	34	34

⑤ この値は35？ → 探索終了

（もし見つからずにここまで到達したら・・・）
配列の末尾？ → 探索終了

線形探索（13_searh.c）

```
//引数nと一致する値を線形探索で見つける
void search(char *label[], int value[], int n, int size)
{
    int i = 0;
    while(1) {
        if(value[i]==n) {
            printf("label: %s, ", label[i]);
            printf("value: %d\n", value[i]);
            break;
        }
        if(i==size-1) {
            printf("見つかりませんでした\n");
            break;
        }
        i++;
    }
}
```

i番目の値はnと一致する？

配列の末尾？

breakで探索を終了する

【練習13-1】

「13_show.c」のサンプルプログラムをコンパイルして、線形構造の出力の実行結果を確認しましょう。

【練習13-2】

「13_search.c」のサンプルプログラムをコンパイルして、線形探索の実行結果を確認しましょう。

【課題13-1】

「13_search.c」のプログラムに対して、
以下のような「ひたちなか市の最低気温」の配列temperature2を作り、線形探索するように変更して下さい。（配列dates1はそのまま使ってよいです）

ひたちなか市の最低気温

7/17	7/18	7/19	7/20	7/21	7/22	7/23
26	26	25	25	25	24	24

【課題13-1】

次のような線形探索をするmainを作って下さい。

① 「最初に24度になるのはいつ？」なのかを線形探索する
② 「最初に20度になるのはいつ？」なのかを線形探索する

[実行結果]

```
label: 7/22, value: 24          (← ①の探索結果)
見つかりませんでした              (← ②の探索結果)
```

【課題13-2】

線形探索search()を参考にして、「引数nよりも小さい最初の値を見つける」関数search_less()を作成してください。

[この関数のプロトタイプ宣言]

```
void search_less(char *label[], int value[], int n, int size);

  /* 探索する条件を「nより小さいか？」に変更する */
```

【課題13-2】

[mainでの処理]

```
  char *dates1[N] = {"7/17", "7/18", "7/19", “7/20",
                     "7/21", "7/22", "7/23"};
  int temperature1[N] = {36, 35, 34, 34, 35, 34, 34};
  search_less(dates1, temperature1, 35, N);
  search_less(dates1, temperature1, 29, N);
```

[実行結果]

```
  label: 7/19, value: 34
  見つかりませんでした
```

【課題13-3】

線形探索search()を参考にして、「引数nと同じ値を全て見つける」関数search_all()を作成してください。

[この関数のプロトタイプ宣言]

```
void search_all(char *label[], int value[], int n, int size);

  /* 「1つでも見つかったかどうか」を表すための変数を用意する */
  /* 見つかった場合は、この変数を1にして、繰り返し処理を続ける */
  /* 配列の末尾になった時に、この変数の値を調べて、
       「1つも見つからなかった」場合のみ「見つかりませんでした」と出力する */
```

【課題13-3】

[mainでの処理]

```
  char *dates1[N] = {"7/17", "7/18", "7/19", “7/20",
                     "7/21", "7/22", "7/23"};
  int temperature1[N] = {36, 35, 34, 34, 35, 34, 34};
  search_all(dates1, temperature1, 34, N);
  search_all(dates1, temperature1, 29, N);
```

[実行結果]

```
  label: 7/19, value: 34
  label: 7/20, value: 34
  label: 7/22, value: 34
  label: 7/23, value: 34
  見つかりませんでした
```

【課題13-4】

線形探索search()を参考にして、「引数nと同じ値を配列の末尾から見つける」関数search_tail()を作成してください。

[この関数のプロトタイプ宣言]

```
void search_tail(char *label[], int value[], int n, int size);

  /* 添字の変数iを配列の末尾から開始するように初期値を与える */
  /* 「配列の末尾？」の条件は「配列の先頭？」に変更する */
  /* 変数iは1つずつ減らしていく */
```

【課題13-4】

[mainでの処理]

```
  char *dates1[N] = {"7/17", "7/18", "7/19", “7/20",
                     "7/21", "7/22", "7/23"};
  int temperature1[N] = {36, 35, 34, 34, 35, 34, 34};
  search_tail(dates1, temperature1, 35, N);
  search_tail(dates1, temperature1, 36, N);
  search_tail(dates1, temperature1, 29, N);
```

[実行結果]

```
  label: 7/21, value: 35
  label: 7/17, value: 36
  見つかりませんでした
```

まだ余裕のある人は…【課題13-5】

課題13-2で作った「引数nより小さい値を見つける」探索に対しても、課題13-3, 13-4のように「全ての値を見つける」「末尾から値を見つける」探索をする関数を作って下さい。

次回は・・・

- データ構造
  - データとは
  - 線形データ構造
- 探索アルゴリズム
  - 線形探索
  - 探索アルゴリズムの高速化
    - 線形探索 ＋ 番兵法
    - 二分探索

小テストについて

小テストの注意点

- □ 他人の力は借りずに、自分だけでプログラムを作成する。（つまり定期試験と同様）

- □ 小テスト中は、演習室外へのネットワークアクセスは遮断される。

小テストについて

小テスト中に参照できるもの

□ **教科書, 配付資料**

□ **自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル**

□ **小テストでは紙媒体のものは参照可能**

□ **上記以外の情報を参照することは不正行為とする**
例：USBで接続された機器に保存されているファイルの参照
ネットワークを介した情報の参照、など