アルゴリズムとデータ構造 / 参考資料

アルゴリズムとデータ構造の参考資料として、用意してみました。各回の授業の課題と併せて、復習にご活用ください。なお、期末試験の試験範囲は、授業で行った範囲すべてです。(この参考資料がすべてではありません。)

【1】アルゴリズムとデータ構造に関する以下の言明が正しいか否かを○と×で答えよ。ただし、とくに説明がない場合は、文中の変数 n は、問 (1)–(5) の問いでは任意の正整数 n を表し、問 (6) 以降の問いではデータ構造に格納された要素数 n を表すと約束する。

(1) $10^3 n + 5n^2 + 100 = O(n)$

(2) $(n+1)(2^{100} + 2m) = O(nm)$

(3) $3.2n + \sin n + \frac{10}{n+1} = O(n)$

(4) $2^n + 4^n + n^3 = O(2^n)$

(5) $2^{2\log n} + 2n^3 = O(n^3)$

(6) 正整数 n が 0 から ∞ に近づく時、関数 $\sqrt{n} = n^{0.5}$ よりも、関数 $\log n$ の方が早く大きくなる。

(7) アルゴリズムは、数学的な性質を形式的に記述したものである。

(8) データ構造は、計算のために記憶領域に効率良くデータを格納するための配置法であり、時間と記憶領域を効率化する。

(9) 抽象データ型は、データ型をそれに適用される一組の操作で抽象的に定めたものである。

(10) 与えられた正整数 a と b の最小公倍数 $gcd(a,b)$ を求めるユークリッドの互除法の時間計算量は、入力値の大きな方の数のビット数の多項式のオーダーをもつ。

(11) 抽象データ型としてみたとき，スタックとキューは，リストの一種である．

(12) 要素の挿入は最後尾、削除は先頭からなされるリストは，待ち行列（queue）であり、LIFO (last-in-fast-out) とも呼ばれる。

(13) 要素の挿入と削除がいつも先頭からなされるリストは，スタック（stack）であり、LIFO(last-in-fast-out) とも呼ばれる。

(14) 双方向連結リストによるリストの実現方法では，任意の正整数 i が指定されたとき、先頭から第 i 番目の場所へのアクセスを $O(1)$ 最悪時間で実現できる。

(15) 双方向連結リスト (doubly-linked list）はポインタで前後の要素の格納領域を指し、一方向連結リスト (singly-linked list) はポインタで次の要素の格納領域を指す。

(16) リストにおいて、挿入演算 INSERT は、指定された値をリストの指定した場所に挿入する演算である。このとき、双方向リストでは、INSERT 演算は $\Theta(n)$ 最悪時間を要する。

(17) リストにおいて、配列によるリストの実現法では，添字によるランダムアクセスができるので，INSERT 演算は $\Theta(1)$ 最悪時間で実現できる。

(18) 待ち行列の主要な応用例の一つとして、ネットワーク通信において到着したデータを保持する有限バッファ（メッセージバッファ）がある。

(19) ヒープの主要な応用例の一つとして、プログラミング言語の処理系における関数呼び出しの実装がある。

(20) ヒープは、全順序集合の要素からなる集合を格納し、任意の要素の挿入（insert）と, 最小要素の削除（deletemin）、探索（search または member）を演算としてもつデータ構造である。

【2】2 分探索木に関する次の問に答えよ。（必須ではないが、手書きの回答の場合は、図を添えると良い。）

(1) 挿入演算 (insert) を用いて、要素列 $S = (4, 2, 6, 1, 3, 5, 7)$ の要素を、空の二分探索木（以降、木と呼ぶ）に、この順に格納して得られる 2 分探索木 T を図示せよ。ただし、空の 2 分探索木からスタートすると仮定する。

(2) 上の (1) で作成した 2 分探索木 T に対して、次の三つの順序による木の巡回 (木のなぞり) を行ったときに、順に出力される要素列を、それぞれ書け。

(a) 前順 (行きがけ順、preorder).

(b) 中順 (通りがけ順、inorder).

(c) 後順 (帰りがけ順、postorder).

(3) 任意の正整数 n に対して、n 個の要素を格納する 2 分探索木の深さが $\Theta(n)$ になるのは、空の木に要素がどのような順番で挿入されるときか答えよ。また、1 から n までの整数を格納するときに、そのような列の一つを具体的に書け。

(4) 空の木に n 個の値をランダムに挿入した場合に，挿入演算の時間計算量 t の期待値 $E[t]$ を、n の関数としてオーダーで与えよ（証明は不要である）。ただし、期待値は、全ての挿入順についてとるものとする。

(5) 節点数 n の平衡探索木において，挿入演算と、削除演算、探索演算を $O(\log n)$ 最悪時間で 実現したい。どのように二分木を改良すれば良いか、その基本的な方針（アイディア）を 1〜3 行程度で簡単に述べよ。

(6) オプション問題。上の (5) で述べた基本的な方針に基づいて、二分探索木における挿入演算と削除演算を $O(\log n)$ 最悪時間で実現するための具体的な方法を説明せよ。最初に、実装の基本方針を述べた後、挿入手続きの説明と計算時間の解析を与えること。削除演算は、簡単に方針を説明するか、説明を省略しても良い。図や疑似コードを用いても良い。（10 行から 1 ページ程度の説明を想定するが、上限は設けない。）

【3】ハッシュ表の二つの実装方法である外部ハッシュ（open hashing, chaining）と内部ハッシュ（closed hashing, open addressing）について、次の問に答えよ。

(1) ハッシュ表は、抽象データ型として、要素の集合 S を格納する辞書構造（dictionary）を実現している。辞書構造がもつ 3 つの主要演算の名称を書き、それらの動作を簡単に説明せよ（各 1 行程度で良い）。

(2) ハッシュ関数 h がもつべき性質を二つ述べよ。さらに、整数を格納する場合に、よく用いられるハッシュ関数の例を与えよ。ただし、例を与える際は、値 x に対するハッシュ関数の値 $h(x)$ を、数式で具体的に書くこと。

(3) ハッシュ表の二つの実装方法である外部ハッシュ（open hashing, chaining）と内部ハッシュ（closed hashing, open addressing）について、入力キー値 x に対するハッシュ値 $h(x)$ の衝突の回避法（衝突対処法）の違いの観点から、それぞれを簡単に説明せよ。

(4) オプション問題。内部ハッシュ法における挿入演算の時間計算量は、ある現実的な仮定のもとで $O(1)$ 平均時間で抑えられる。そのことを、理由を添えて説明せよ。（10 行から 1/2 ページ程度の説明を想定するが、上限は設けない。）

【4】再帰アルゴリズムに関する以下の問いに答えよ。

(1) ある再帰アルゴリズムについて、入力 n に対する時間計算量を解析したところ、$T(1) = 1$, $T(n) = 2T(n-1)$ と分かった。この時、$T(n)$ を n の関数として表し、そのオーダーを示しなさい。

(2) ある再帰アルゴリズムについて、入力 n に対する時間計算量を解析したところ、$T(1) = 0$, $T(n) \leq T(n/3) + c$ と分かった。ここで、c は定数である。この時、$T(n)$ の上界を n の関数で表し、そのオーダーを示しなさい。

(3) ある再帰アルゴリズムについて、入力 n に対する時間計算量を解析したところ、$T(1) = 0$, $T(n) \leq 3T(n/3) + c$ と分かった。ここで、c は定数である。この時、$T(n)$ の上界を n の関数で表し、そのオーダーを示しなさい。

(4) オプション問題。非負整数 n, m に対し、以下のように $f(n,m)$ を定義する。この時、$f(13, 10)$ の値を求めなさい。

$$f(n,m) = \begin{cases} n & (m = 0 \text{ の時}) \\ f(n/2, m-1) & (m > 0 \text{ かつ } n \text{ が偶数の時}) \\ f(3n+1, m-1) & (m > 0 \text{ かつ } n \text{ が奇数の時}) \end{cases}$$

手で計算しても、プログラムを作成して計算しても構わない。手で計算した場合には、その過程も、プログラムを作成した場合には、そのプログラムも示すこと。

(5) オプション問題。上記 (4) の $f(n,m)$ について観察し、m が 10 以上の整数の場合に $f(13, m)$ がどのような値を取るかを示しなさい。

【5】ソートに関する以下の問いに答えよ。

(1) 第 9 回の講義資料に記載された選択ソートのアルゴリズムの実行途中で、`A[0]` 〜 `A[9]` は 0, 1, 3, 2, 5, 8, 5, 7, 5, 6 で、$\texttt{n} = 10$, $\texttt{i} = 2$ とする。ここで、 Step 2 で

$$\texttt{j} \leftarrow \arg\min\{\texttt{A}[\texttt{j}] : \texttt{i} \leq \texttt{j} < \texttt{n}\}$$

を実行した際の `j` の値を求めなさい。

(2) 第 10 回の講義資料に記載されたヒープソートのアルゴリズムを実行するために、配列にヒープを格納したい。`A[0]` 〜 `A[9]` が 7, 25, 30, 21, 20, 28, 20, 19, 20, 10 の時に $\texttt{HEAPIFY}(\texttt{A}, 0, 10)$ を実行する過程を、配列と 2 分木を対応させながら説明しなさい。

(3) オプション問題。ヒープソートのアルゴリズムにおいて、大きさ n の配列に対して `HEAPIFY` を 1 回実行するのに必要な計算時間の上界を、その根拠とともにオーダーで示しなさい。

【6】グラフに関する以下の問いに答えよ。

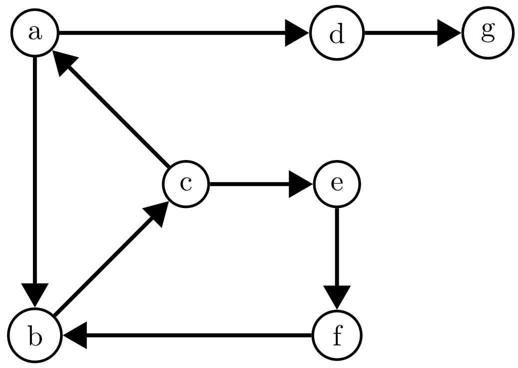
(1) 有向グラフ $G = (V, A)$, $V = \{v_0, v_1, v_2, v_3, v_4, v_5\}$, $A = \{(v_0, v_1), (v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_0), (v_1, v_3)\}$ を図に示しなさい。また、この有向グラフの隣接行列を求めなさい。

(2) 次の表の隣接リスト表現で与えられる有向グラフ G を図に示しなさい。

頂点	隣接リスト
a	b, f, g
b	d, c, f, h
c	d, e
d	null
e	null
f	null
g	null
h	g

(3) 無向グラフ $G = (V, E)$ に対し、$(\sum_{v \in V} \deg_G(v))/|V|$ すなわち「各頂点の次数の総和 / 頂点数」を平均次数という。完全二部グラフ $K_{m,n}$ の平均次数を、m, n に関して求めなさい。

(4) 以下に示す図の有向グラフに対し、授業スライドの強連結成分分解を行う。各頂点において、その隣接リストはアルファベット順に並んでいるとし、頂点 a から Step 1 を始める。この時、Step 1 の終了時点での各頂点の `f[ ]` の値を示しなさい。(有向グラフを描き、各頂点の横に `f[ ]` の値を示しなさい。)



(5) 上記の Step 1 に引き続き、Step 2 では、ある順番に従って頂点を選択して DFS を実行することを繰り返す。<u>どの頂点から</u> DFS を実行して、<u>どの連結成分</u> が得られるかを、実行の順に説明しなさい。なお、1 つの強連結成分はその頂点の集合で表す。また、記述の詳しさは、授業スライドの説明程度で構わない。

(6) オプション問題。頂点数 n、辺の本数 m の有向グラフに対して強連結成分分解を行う。取りうる強連結成分の個数の最大値と最小値を答えなさい。また、$n = 6$ の場合において、強連結成分の個数がその最大値となるようなグラフの例と、その最小値となるようなグラフの例を、それぞれ 1 つずつ示しなさい。