

# データ構造とアルゴリズム

問題 1. 以下の C 言語プログラムは、マージソートのアルゴリズムによって、与えられた配列  $a[]$  の要素を昇順に並べ替えるものである。以下の設問について、指定された答案用紙に解答しなさい。

```

/**
 * マージソート. 配列 a[] のうち a[p] ~ a[p+n-1] をソートする. パラメータは以下の通り.
 * a[]: ソート対象の配列
 * b[]: マージ操作に使用する作業用配列
 * p: ソート対象とする a[] の部分配列の先頭インデクス
 * n: ソート対象とする a[] の部分配列の要素数
 */
void msort(int a[], int b[], int p, int n) {
    int h;
    if (n > 1) { /* 要素数が 1 より大きい場合は並べ替えを実施 */
        h = n/2; /* n を半分に割って */
        msort(a, b, p, h); /* 前半部分をソート */
        msort(a, b, ①); /* 後半部分をソート */
        merge(a, b, ②); /* マージする */
    }
}
/**
 * マージする. 配列 a[] のうち, 【部分 1】 a[p] ~ a[n/2-1] と 【部分 2】 a[n/2] ~ a[p+n-1] を比較
 * しながら昇順に並ぶように結合する. 部分 1, 2 はソート済. パラメータの意味は msort() に同じ.
 */
void merge(int a[], int b[], int p, int n) {
    int i, j, k, h;
    h = n/2; /* n の半分 */
    i = p; /* 部分 1 の走査用 */
    j = p+h; /* 部分 2 の走査用 */
    /* 部分 1 と部分 2 の要素を前から 1 つずつ比較しながらマージしていく */
    for (k=p; k<p+n; k++) {
        /* 部分 2 を最後まで走査した, または, 部分 1 を走査中で部分 1 の要素がより小さい */
        if ((j == p+n) || ③) {
            /* 部分 1 の要素を作業用配列 b[] に取り出し, i を進める */
            b[k] = a[i]; i++;
        } else {
            /* 部分 2 の要素を作業用配列 b[] に取り出し, j を進める */
            b[k] = a[j]; j++;
        }
    }
    /* 結合された配列 b[] を a[] に上書きする */
    for (k=p; k<p+n; k++) {
        ④;
    }
}

```

設問 (1) プログラム中の空欄①～④を埋めなさい。

設問 (2)  $a[] = \{8, 1, 4, 20, 21, 18, 27, 15\}$  に対して,  $\text{msort}(a, b, 0, 8)$  を実行した際,  $a[]$  がどのように分割されマージされていくかを, 答案用紙の空白部分を補うように図示しなさい。

設問 (3)  $n$  をソートする配列の要素数,  $T(n)$  をマージソートに要する時間とする. このとき,  $T(n)$  の時間計算量が  $O(n \log n)$  であることを示しなさい. 簡単のため,  $n=2^k$  としてよい。

設問 (4) コンピュータのメモリに一度に載りきらないような巨大なデータでも, マージソートの仕組みを使えば (原理的に) ソートすることが可能である. その具体的な方法について述べなさい。

問題 2. 英単語 (すべて小文字からなる) の集合を辞書順\*に保持する二分探索木 Tword を考える. Tword では, 各ノードは単語 1 語に対応し, 左の子ノードは辞書順で先に来る単語, 右の子ノードは辞書順で後に来る単語を保持する. この時, 以下の設問について, 指定された答案用紙に解答しなさい.

設問(1) Tword が空の状態のとき, 次の単語を左から順に 1 つずつ Tword に挿入した場合, 最終的に得られる Tword の形を図示しなさい.

hat sun map bat jam yet cat pen

設問(2) (1) で得られた Tword を使って, pen を検索する方法を説明しなさい.

設問(3) (1) で挿入したすべての単語を辞書順に表示するには, Tword をどのように探索し, どのタイミングで単語を表示すればよいか答えなさい.

設問(4) (1) で得られた Tword から sun を削除した時, 得られる Tword の形を図示しなさい. 続けて pen を削除した時, 得られる Tword の形を図示しなさい (2 つの Tword を描くこと).

設問(5) 二分探索木は与えられた要素が集合内に存在するかどうかを高速に探索できるデータ構造である. しかし, 要素の挿入順序によっては, 効率的な検索ができない構造になってしまう. そのような例を 1 つ挙げ, その解決方法を示しなさい.

【\*辞書順】辞書に単語が載っている順番のように, 文字や文字列を並べる順序のこと. 英単語の場合, まず先頭の文字を見て, アルファベット順 (a, b, c, d, …) に並べる, 先頭の文字が同じ場合には, 2 文字目を見て順序をつける. 2 文字目も同じ場合, 3 文字目... 以下同様.