

早稲田 2019 年度解答例

文殊の知恵
中田昌輝

問題番号 1

科目名 情報基礎

問題番号 1 問題文

以下の問 (1)～(3) に答えよ。

(1) 集合 $\Sigma = \{0, 1\}$ について, Σ 上の記号列全体の集合 (the set of all strings over Σ) を Σ^* で表す。

(1-a) 次の文が正しい定義となるように空欄【 】を埋めよ。

Σ 上の言語 (language over Σ) とは Σ^* の【 】のことである。

(1-b) Σ 上の言語 A, B が等しいとはどういうことか書け。

(1-c) 言語 $\{0^{2n}1 | n \geq 1\}$ を表現する正則表現 (regular expression) を書け。

(1-d) 次の文が「言語 $L = \{0^n 1^n | n \geq 1\}$ は正則言語ではない」ことの証明となるように, 空欄【 】を埋めよ。

L を認識 (recognize) する決定性有限オートマトン (deterministic finite automaton) の状態 (state) の数を m とすると, 【 】が受理 (accept) される過程でこのオートマトンが入る最初の $m + 1$ 個の状態 $q_0, q_1, q_2, \dots, q_m$ のうち 2 つは同じ状態だから, それらを $q_i, q_j (0 \leq i < j \leq m)$ とすると, $0^{m-(j-i)}1^m \in L$ となり矛盾する。

(2) 記号表 (symbol table) とは, キー (key) を持つ項目 (item) からなるデータ構造 (data structure) であり, 次の 2 つの基本操作 (basic operations) を持つ。

新しい項目を挿入 (insert) する

与えられたキーを持つ項目を探索 (search) する

次の表は, 記号表における挿入コストと探索コストを表したものである。ここで, 記号表中の項目の個数を N とし, コストとは N の関数としておおよその計算時間を表している。空欄 (a)～(e) にもっともふさわしいものを, 下の選択肢 (ア)～(オ) から選べ。同じ選択肢を複数回選んでもよい。

実装方法	最悪の場合		平均の場合		
	挿入	挿入	挿入	成功探索	不成功探索
逐次探索	N	(a)	$N/2$	$N/2$	(b)
2 分探索木	(c)	N	$\log_2 N$	(d)	(e)

選択肢: (ア) 1 (イ) N (ウ) $N/2$ (エ) $\log_2 N$ (オ) $N \log_2 N$

(3) 右図のプログラム片 (code fragment) を考える。関数 (function) $\text{sort1}(L,R)$, $\text{sort2}(L,R)$, $\text{sort3}(L,R)$ はいずれも、グローバルスコープの配列 A に格納されている値 $A[L], \dots, A[R]$ を昇順 (ascending order) に整列 (sort) をすることをめざしている。ただし引数 L,R が $L < R$ を満たし、 $A[L], \dots, A[R]$ が A の定義範囲内となるように呼ばれるものとする。以下、これら 3 関数の整列と呼ぶ。

(3-a) 各整列関数の目的を熟考して、正しく完成させるため、右図中の空欄【あ】、【い】、【う】を適切に埋めよ。

(3-b) 3 つの整列関数が正しく完成したとき、最も早いものの漸近時間計算量 (asymptotic time complexity) は何か。整列する要素数を n とせよ。

(3-c) 3 つの整列関数が正しく完成したとき、そのうち 1 つは、すでにほとんど昇順に整列している入力列について他の 2 つより速い。その関数はどれか。

(3-d) 3 つの整列関数のうち、正しく完成したとき安定 (stable) であるものをすべて答えよ。

(3-e) 右図の 3 つの整列関数のうち 1 つは誤りを含み、 $A[L], \dots, A[R]$ の範囲外の配列を参照してしまう。

i) 誤りのある箇所の行番号 (line number) を答えよ。

ii) 誤りを含む整列関数を実行した場合、最初に範囲外の配列要素を参照する箇所の行番号を答えよ。

```

1 fun exch(i,j)
2   t ← A[i]
3   A[i] ← A[j]
4   A[j] ← t
5 end fun
6
7 fun compexch(i,j)
8   if (A[j] < A[i])
9     exch(i,j)
10  end if
11 end fun
12
13 fun sort1(L, R)
14   i ← L
15   while (i < R)
16     min ← i
17     j ← i + 1
18     while(j ≤ R)
19       if (A[j] < 【あ】)
20         min ← j
21       end if
22       j ← j + 1
23     end while
24     exch(i, min)
25     i ← i + 1
26   end while
27 end fun
28
29 fun sort2(L, R)
30   i ← L
31   while (i < R)
32     j ← R
33     while(j > i)
34       compexch(j-1, 【い】)
35       j ← j - 1
36     end while
37     i ← i + 1
38   end while
39 end fun
40
41 fun sort3(L, R)
42   i ← R
43   while(i ≥ L)
44     compexch(i-1, i)
45     i ← i - 1
46   end while
47   i ← L + 2
48   while(i ≤ R)
49     j ← 【う】
50     v ← A[i]
51     while(v < A[j-1])
52       A[j] ← A[j-1]
53       j ← j - 1
54     end while
55     A[j] ← v
56     i ← i + 1
57   end while
58 end fun

```

問題番号 1 解答例

(1) 情報数学の分野である.

(1-a) Σ 上の言語とは Σ^* からある特定の条件を満足する記号列だけを集めた集合であるので空欄に入れるべき語句は部分集合である.

(1-b) Σ 上の言語 A, B が等しいとは同じ記号列の集合であるということであるので,
言語 A のみを受理するオートマトンと言語 B のみを受理するオートマトンが等価であるということである.

(1-c) 0 が偶数回並んだあと, 1 が来る文字列を正則表現に変えればいいので, $(00)^*1$

(1-d) この言語 L のみを受理するオートマトンが存在しないということである. これはオートマトンが何文字同じ文字を読み込んだかをカウントする機能がないためである (特定の文字数をカウントすることは可能である). したがって n 回 0 を読み込んだと記憶を保持することができないので, 1 を n 回読み込むようにすることは不可能である. これを証明すると同じ状態が複数回出てしまうことを用いて背理法で示す. よって, 空欄に入れるべき語句は 0^m1^m である.

(2) アルゴリズムとデータ構造の分野である.

(a) 逐次探索は順番に配列を参照していくが探索において最悪の場合は目的のものが最後に存在する場合であるので N , つまり(イ).

(b) 整列した配列を使うので, 目的の値が飛ばされていたら不成功であると判断できるので成功探索と同様に $N/2$, つまり(ウ).

(c) 2 分探索木で最悪の場合は挿入された順番が整列された値であった場合である. これは 2 分木にならず右肩下がりまたは右肩上がりどちらかの木になってしまう. このあと挿入されるとき一番根から離れた場所に挿入する (一番子どもの節点) 場合は特に最悪で, すべての要素を見ることとなるので N , つまり(イ).

(d) 平均の場合は平衡状態が取れている二分木探索と捉えることができる. このとき成功探索は高さが $\log_2 N$ となるので $\log_2 N$, つまり(エ).

(e) 不成功探索も成功探索と同様に考えることができるので, $\log_2 N$, つまり(エ).

(3) `exch` 関数について, これはよく見る `swap` 関数である. つまり配列の添字を付け替える, 値を入れ替える関数である. つぎに `compexch` は値の大小が逆であった場合は `exch` 関数を呼び出して値の整列を行っている.

(3-a) `sort1` 関数についてみる. 配列の添字を参照するための変数 i を左端つまり一番小さいと思われる配列の添字に代入している. これを左から右まで順に動かす `while` 文がみえる. 一旦 i を別のところ (`min`) に格納し, j に i の次の添字を格納している. その次に j が右端になるまで比較を行っている. ここで最小値を見つけ次第 j の値を `min` にいれて最小値を探していると判断することができる. この最小値が入っている添字は本来, i のところにあるべきであることを考慮すれば【あ】に入るべき語句は $A[\text{min}]$ であるとわかる. これは選択ソートである.

つぎに `sort2` 関数についてみる. i を左端に, つぎに i が右に行くまで j を右端にする `while` 文がみえる. $j > i$ とつねに i は j 以下という条件で 2 つめの `while` 文を回している. `while` 文内で j を 1 ずつ減らしていることから i を基準とし, j を右から順々にずらしていることがわかる. つまり左端から順にソートされていくことがわかる. ここで `compexch` 関数は大小を比較して入れ替えるというものであった. つまり i のほうが $j-1$ より小さいべきであり j のほうが $j-1$ より大きいべきである. ここでどちらがふさわしいかを考える. i と $j-1$ を入れ替えるソート方法であると不安定になってしまうので, 安定にソートする方法としては j と $j-1$ を入れ替えるソートである. 【い】に入れるべき語句は j であるとわかる. これはバブルソートである. 最後に `sort3` 関数についてみる. i を右からみていき左端になるまで一つ前の $i-1$ と入れ替えていることがわかる. これからいえることとしては左端の値がソートされた状態になるということである. つぎに i

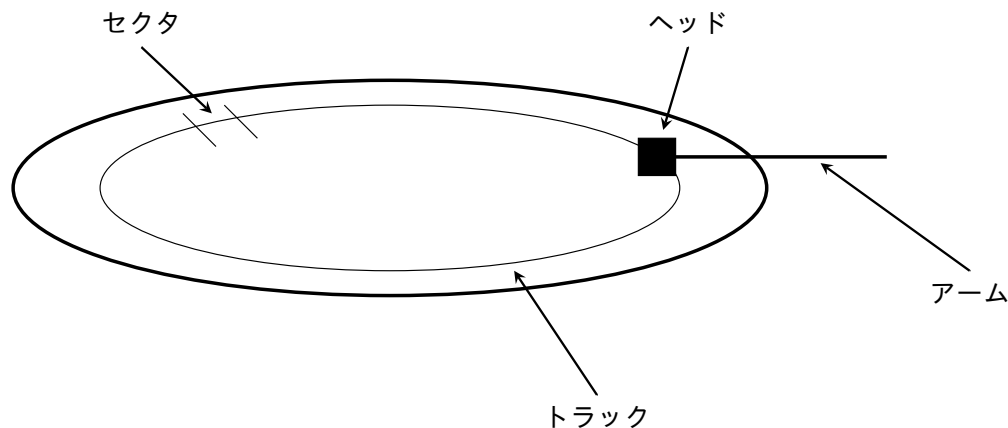
に $L+2$ を入れた式からみると i が右端に行くまで j の値を変えている。つまり i を左から揃えていくためのソートされた列か未ソートの列かを区別するための基準となる変数であることがわかる。つまり j を右端から揃えていき、未ソートの列の左端 $A[i]$ を v にいれることで一つずつ揃えていることが分かる。ゆえに【う】に入れるべき語句はRであることがわかる。これは挿入ソートである。

- (3-b) どのソートの計算量も $O(n^2)$ である。平均として $n(n-1)/2$ の比較を行う。バブルソートの場合はどういう状況においても $n(n-1)/2$ の比較を行う。
- (3-c) すでにほとんど昇順に整列している入力列については挿入ソートは $O(n)$ の速さでソートすることが可能である。というのも小さい値が見つかった場合は挿入ソートの場合は配列の添字を一つずつずらして値を挿入するという操作が行われるが、整列されている場合はその操作がなくなるので早く実行することができる。つまり答えはsort3である。
- (3-d) 安定であるかどうかについて、安定であるものは $1, 3, 2_A, 2_B, 4$ というソートに対し、 $1, 2_A, 2_B, 3, 4$ というふうにいかなる状況でもソートされるものを指す。一方で不安定であるものとは $1, 2_B, 2_A, 3, 4$ というふうに同じ 2 という数字でも入れ替えが起こってしまうことをいう。この不安定な状況が起こるのは隣同士の値を比較するのではなく離れた場所での値を比較するために起こる。ゆえにバブルソートは隣同士の値を比較するため安定である。また挿入ソートも後方から値を確認していくため、同じ値におけるソートの入れ替えが起きない。一方で選択ソートは最小値をとりそれを入れ替えるという方法でソートしていくので離れた場所での値を比較することとなる。そのため安定ではない。ゆえに答えはsort2, sort3である。
- (3-e) 見るべきポイントとしては加算や減算処理が行われているときのループ文である。
- i) $i \geq L$ というのをみるとそのあと $i-1$ を関数の引数にとっているので $L-1$ が参照されることとなる。ゆえに43行。
 - ii) 関数が実行されるときなので44行。

問題番号 2

科目名 計算機システム

問題番号 2 問題文



磁気ディスク (Magnetic Disk) とは、上図に示すようなデータ記憶に磁性体を塗布した円盤を回転させておこなう記憶媒体の名称である。磁気ディスク上のデータの読み書きに関する以下の設問に解答せよ。各円盤上のトラック (Track) 数を 30, 1 つのトラック内のセクタ (Sector) 数を 100, 円盤は 1ms で 1 回転し, アーム (Arm) に取り付けられたヘッド (Head) の移動時間に関してはトラック間の移動に 2ms 必要とするとする。また, 各セクタは 64 バイトのデータを格納できるとする。

- (1) 指定されたセクタを読み出す要求があった場合, そのセクタの読み出しに必要な最悪遅延時間を示せ。その際, 最悪遅延時間の結果の導出過程を示すこと。
- (2) 1 つの円盤上の各トラック名を T1-T30 と表現する。今, ホスト計算機から, T20, T5, T25, T13, T1, T19 の順にデータの読み込み要求があったとする。この場合, FCFS(First Come First Served) スケジューリングを用いてトラックを読み込んだ場合と SCAN スケジューリングを用いて読み込んだ場合のそれぞれのトータルの読み込みに必要な時間を示せ。各トラック内のセクタのアクセス時間に関しては最悪の読み込み時間を想定して計算すること。また, 結果の導出過程に関しても記載すること。初期状態として, ヘッドは T1 の位置に移動しているとする。
- (3) バッファキャッシュ (Buffer Cache) は磁気ディスク内の頻繁に利用するデータを実メモリ内に格納するオペレーティングシステムの機能である。今, 128 バイトのデータを読み込む場合, すべてのデータがバッファキャッシュ上に格納できている場合と, 全く存在しない場合の最悪の読み込み時間を示せ。結果の導出過程に関しても記載すること。ここでは, バッファキャッシュのブロックサイズは 64 バイトとし, 実メモリ 1 バイトの読み込み/書き出し時間を 1μ 秒とする。また, バッファキャッシュ上にデータが存在しない場合は, 磁気ディスクからデータを読み出してメモリに格納するものとする。
- (4) 想定するバッファキャッシュは 5 ブロック分のメモリが割り当てられるものとする。ここでは, それぞれのブロックを B1-B5 として表現する。また, あるトラック上のセクタ S1-S30 の一部を読み込む場合を考える。今, S1, S2, S3, S4, S1, S2, S5, S6, S7, S1, S2, S3 の順にセクタが読み込まれる場合を考える。この場合, バッファキャッシュを管理するポリシーとして FIFO(First In First Out) を用いた場合と, LRU(Least Recently Used) を用いた場合で, B1-B5 が格納するデータが格納されたセクタ番号がどう変化するかを時系列順に図示せよ。

問題番号 2 解答例

- (1) 指定されたセクタを読み出す要求があった際、ヘッドが目的のトラックにいない場合はトラック間の移動が 2ms であるから最悪の場合は 29 回トラック間の移動が発生するので $2 \times 29 = 58\text{ms}$ にかかることとなる。次に目的のトラックにいる場合でもセクタが目的の場所ではない場合は円盤が回転する必要があるので、最悪の場合は 1 回転する必要があるため求める答えは

$$58 + 1 = 59\text{ms}$$

- (2) 各トラック内のセクタのアクセス時間は最悪の読み込み時間を想定するのでセクタの移動は毎回 1ms かかる。FCFS スケジューリングを用いるので、T1 から T20 への移動は 19 回移動する。T20 から T5 は 15 回移動し、T5 から T25 は 20 回、T25 から T13 は 12 回、T13 から T1 は 12 回、T1 から T19 は 18 回の移動となるので合計で $(19 + 15 + 20 + 12 + 12 + 18) = 96$ 回移動することとなる。また各トラックで 1 回転すると考えれば 6 回転するので求める答えは

$$96 \times 2 + 6 \times 1 = 198\text{ms}$$

となる。

- (3) すべてのデータがバッファキャッシュ上に格納できている場合はメモリの読み込みだけで済むので、 128μ 秒である。一方で全く存在しない場合の最悪の読み込み時間はヘッドの移動をして読み込みを行いメモリに書き込む作業が 2 回行われるので、それにかかる時間は

$$2 \times (2 \times 29 + 1) = 118\text{ms}$$

読み込みは書き出して読み込むので合計で 4μ 秒かかるとすれば合計で

$$118\text{m} + 4\mu = 118.128\text{ms}$$

となる。

- (4) fdfd