

# 解答・解説編

## Part 1

テクノロジー系 ..... 40

## Chapter 1 基礎理論

## 演習1-1

→別冊テキストP.2

## 解答-エ

選ばれなかったものについて解説をしておきます。

ア. 経路長が1以下の組合せは、(1,0)、(-1,0)、(0,1)、(0,-1)、(0,0)の5つがあります。

イ.  $d(X_1, Y_1, X_2, Y_2) = d(0, 0, 1, 0)$  のように、行方向あるいは列方向の座標が等しい場合、条件を満たしません。

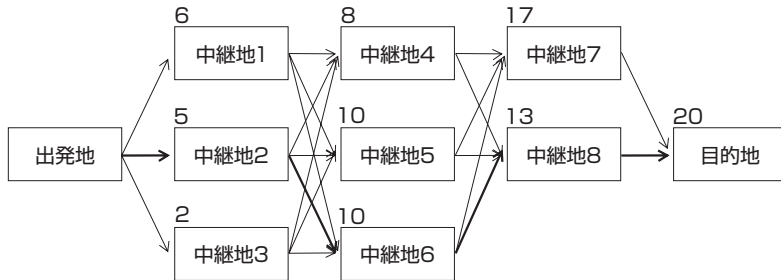
ウ.  $X_1, X_2$  が等しく、かつ、 $Y_1, Y_2$  が等しい場合に、経路長は0となります。このとき、X座標とY座標の値が等しくなくてもかまいません。

## 演習1-2

→別冊テキストP.2

## 解答-イ

各経路をつなぐ運賃のうち、そこまでの累積が最も少ないものを選択しながら、目的地まで到達するように経路を選択します。



各目的地地上にある数値は、そこまでの累積運賃のうち一番低いものであり、太線の矢印が、目的地までの累積運賃が最小の経路です。

上図より、解答はイとなります。

## 演習1-3

→別冊テキストP.8

## 解答-イ

$$\begin{aligned}
 \text{平均ビット数} &= 1 \times 0.5 + 2 \times 0.3 + 3 \times 0.1 + 4 \times 0.05 + 4 \times 0.05 \\
 &= 0.5 + 0.6 + 0.3 + 0.2 + 0.2 \\
 &= 1.8
 \end{aligned}$$

## 演習1-4

→別冊テキストP.8

### 解答-ア

平均ビット数

$$\begin{aligned} &= 1 \times 0.4 + 2 \times 0.19 + 4 \times 0.1 + (5 \times 0.05) \times 5 + (6 \times 0.03) \times 2 \\ &= 0.4 + 0.38 + 0.4 + 1.25 + 0.36 \\ &= 2.79 \\ &\approx 2.8 \end{aligned}$$

## 演習1-5

→別冊テキストP.11

### 解答-エ

例えば、

aaaabccdddddeee

という1文字1バイトで表現される15バイトの文字列に対し、

4a1b2c5d3e

のように反復回数と文字の組に置き換えて表現すると10バイトとなり、文字列を短縮することができます。これをランレングス法と呼び、データの圧縮に利用されます。

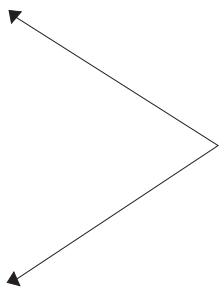
なお、この方法では、文字の繰返し回数が少ない場合は、変換後の文字列長が元の文字列よりも長くなってしまうことがあります。

## 演習1-6

→別冊テキストP.13

## 解答-ウ

ハッシュ関数として、 $f(\text{データ}) = \text{mod}(\text{データ}, 8)$  が与えられているため、この関数を使って格納位置を求めます。データが16進数で表現されているため、16進数を2進数に置き換えてから解答を求めることにします。また、データを8 ( $2^3$ ) で割ることから、3ビット右にシフトします。シフトした結果、アンダフローした値が余りとなり、格納アドレスとなります。順に示すと下記ようになります。

	8で割った余り(アンダフロー)
①(1A) <sub>16</sub> = (00011010) <sub>2</sub> … (010) <sub>2</sub>	
②(35) <sub>16</sub> = (00110101) <sub>2</sub> … (101) <sub>2</sub>	
③(3B) <sub>16</sub> = (00111011) <sub>2</sub> … (011) <sub>2</sub>	
④(54) <sub>16</sub> = (01010100) <sub>2</sub> … (100) <sub>2</sub>	
⑤(8E) <sub>16</sub> = (10001110) <sub>2</sub> … (110) <sub>2</sub>	
⑥(A1) <sub>16</sub> = (10100001) <sub>2</sub> … (001) <sub>2</sub>	
⑦(AF) <sub>16</sub> = (10101111) <sub>2</sub> … (111) <sub>2</sub>	
⑧(B2) <sub>16</sub> = (10110010) <sub>2</sub> … (010) <sub>2</sub>	
⑨(B3) <sub>16</sub> = (10110011) <sub>2</sub> … (011) <sub>2</sub>	

以上のことから、⑧で初めて衝突が発生するため、格納できないデータはB2となります。

## 演習1-7

→別冊テキストP.13

## 解答-ア

ハッシュ法は、記録されたレコードを探索するアルゴリズムの1つです。ハッシュ関数を用いてキー値に演算を施し、記録位置を求めます。高速な探索方法ですが、キー値によっては同じ演算結果が出てしまい、格納場所の衝突が発生してしまうという欠点があります。また、最初に求めた記録位置に格納できたレコードをホームレコードといい、ハッシュ関数を用いて求めた結果、すでに他のデータが格納されている場所が記録位置になってしまったレコードをシノニムレコードといいます。

## 演習1-8

→別冊テキストP.13

### 解答-ウ

ハッシュ法は、ハッシュ関数と呼ばれる計算式を用いてキー値からデータの格納アドレスを求める探索法であり、目的のデータがほぼ1回で探索できるという特徴をもっています。ただし、ハッシュ法では、ハッシュ関数によりキー値をある一定の範囲にまとめるため、異なるキー値から、同一の格納場所が求められることがあります。この現象を衝突（コリジョン）といい、衝突が発生したときのために、チェーン法やオープンアドレス法などの格納方法が用いられています。

選ばれなかったものについて解説をしておきます。

- ア. ハッシュ法では、2分木は使用しません。
- イ. ハッシュ関数で得た値によっては、衝突が起こる可能性があります。
- エ. 衝突が起こらない限り、一意に格納アドレスを得ることができるので、探索時間は一定です。

## 演習1-9

→別冊テキストP.13

### 解答-イ

ハッシュ値 $h$ を求める計算式が問題文中にあるので、まず、キー“SEP”の値を求め、それと同値のキーを探します。

SEPのハッシュ値

$$\text{キーの各アルファベットの順位の総和} = 19 + 5 + 16 = 40$$

$$h = 40 \bmod 27 = 13$$

解答群のハッシュ値

$$\text{APR} = 1 + 16 + 18 = 35 \quad h = 35 \bmod 27 = 8$$

$$\text{FEB} = 6 + 5 + 2 = 13 \quad h = 13 \bmod 27 = 13 \quad \leftarrow \text{衝突する}$$

$$\text{JAN} = 10 + 1 + 14 = 25 \quad h = 25 \bmod 27 = 25$$

$$\text{NOV} = 14 + 15 + 22 = 51 \quad h = 51 \bmod 27 = 24$$

## 演習1-10

→別冊テキストP.13

### 解答-エ

4,999を除数とする剰余を用いるハッシュ関数で、キー値が12,500のレコードに対するシノニムレコードのキー値を求めると、 $12,500 \div 4,999 = 2$ あまり2502から2502となります。

同様の方法で各選択肢のキー値を求めると、次のようになります。

- ア.  $2 \div 4,999 = 0$ あまり2
- イ.  $1,250 \div 4,999 = 0$ あまり1,250
- ウ.  $2,501 \div 4,999 = 0$ あまり2,501
- エ.  $17,499 \div 4,999 = 3$ あまり2,502

## 演習1-11

→別冊テキストP.13

### 解答-ウ

この問題においてのハッシュ値の求め方は、キー値を10で割ったときの余りであるので、余りとしては0から9の10とおり存在します。このとき、3つのデータを選んで衝突の起こらない確率を求めればよく、

$$10/10 \times 9/10 \times 8/10 = 0.72$$

となります。

## 演習1-12

→別冊テキストP.13

### 解答-イ

一般に、キー値の分布によらず、できるだけ均等で、かつ広範囲なハッシュ値を与えるハッシュ関数が“よい”ハッシュ関数といえます。

本問におけるハッシュ値の範囲を求めると、以下のようになります。

ア. 0~255

イ. 1~16

ウ. 0~255

エ. 0~255

したがって、イはハッシュ関数として不適切です。

## 演習1-13

→別冊テキストP.13

### 解答-イ

ハッシュ関数で求められる値(ハッシュ値)が等しいとき、衝突が起きます。キーがaであるデータと、キーがbであるデータのハッシュ値は等しくなり、それをrとすれば、 $a \bmod n = b \bmod n = r$  となります。

aをnで割ったときの商を $q_a$ 、bをnで割ったときの商を $q_b$ とすると、次の式が成り立ちます。( $q_a, q_b$ は整数)

$$a = n \times q_a + r \cdots \text{①}$$

$$b = n \times q_b + r \cdots \text{②}$$

次に①-②を求めると、

$$a - b = n(q_a - q_b)$$

$q_a, q_b$ は整数であることから、 $q_a - q_b$ も整数となります。

したがって、 $a - b$ はnの整数倍であることがわかります。

## 演習1-14

→別冊テキストP.13

### 解答-ア

- ①キー値「55550」を11進数と見なし、10進数に変換する。  
 $5 \times 11^4 + 5 \times 11^3 + 5 \times 11^2 + 5 \times 11^1 + 0 \times 11^0 = 80520$
- ②80520の下4けたに対して0.5を乗じる。  
 $520 \times 0.5 = 260$   
したがって、4けたで表現すると0260となります。

## 演習1-15

→別冊テキストP.13

### 解答-イ

データ件数を $n$ とすると、線形探索の平均比較回数は $n/2$ 回、2分探索の平均比較回数は $\log_2 n$ 回となることを理解しておくことが必要です。

- ア. 2分探索木を用いた探索の平均比較回数は、2分探索と同じ $\log_2 n$ 回です。
- イ. ハッシュ表を用いた探索は、探索対象のキー値でハッシュ関数の値を計算した結果が、探索対象のデータの場所を示します。このとき、複数のキー値から同じハッシュ関数の値が求まることもあり、そのことを衝突と呼びます。衝突が発生した場合、新たな探索が必要になりますが、ここでは、衝突を無視してよいので、新たな探索は必要ありません。したがって、最初の1回の計算で、探索できることになり、平均比較回数は1回となります。
- ウ. 2分探索の平均比較回数は $\log_2 n$ 回です。なお、2分探索は整列済みの配列でしか処理できません。
- エ. 重複登録のないリストを用いた線形探索は、先頭から順番に探索をしていくため、平均比較回数は $n/2$ 回となります。

## 演習1-16

→別冊テキストP.13

### 解答-ア

探索手法の特徴に、各探索表の構成法が当てはまるかどうかを考えていけば、正しい組合せが導き出せます。

2分探索には、データがキー値の昇順または降順で整列されていなければならないという前提条件があります。3つの中でキー値の順に整列されているのはaしかなく、したがって、aにふさわしい探索手法は2分探索です。

ハッシュ表探索は、ハッシュ関数を用いてキー値を計算し、そこから格納場所を求める探索法です。衝突が起こらない、という前提のもとであれば、その格納場所はデータに対して一意となります。したがって、cの「コードから一意に決まる場所に格納した探索表」という構成法から、ふさわしい探索手法はハッシュ表探索です。

以上から、必然的にbの構成法にふさわしい探索手法は、線形探索となります。

## 演習1-17

→別冊テキストP.13

### 解答-ア

O(オーガ)記法は、定数や係数を除外して、最も増加率の大きいnの項だけで表します。  
各探索法の実行時間のオーガを示すと、次のようになります。

- ・線形探索  $n$
- ・2分探索  $\log_2 n$
- ・ハッシュ探索(衝突が起こらない場合)  $1$

今回の問題は、衝突する確率は無視できるほど小さいとなっているため、衝突が発生しないと考えることができます。したがって、探索する場合にハッシュ値の計算は1回で済ませることができます。

したがって、解答はアとなります。

## 演習1-18

→別冊テキストP.13

### 解答-エ

探索のアルゴリズムについてのグラフです。それぞれの特徴を整理します。

- ア. 指数関数的に増加していくことから、整列(バブルソートなど)のグラフを表しているものと考えられます。
- イ. 一次関数(直線)的に増加していることから、線形探索のグラフを示しています。
- ウ. 対数関数的に増加していることから、2分探索のグラフを示しています。
- エ. データの個数に依存せず常に一定の時間であることから、ハッシュなどインデックスを使った探索のグラフを表しているものと考えられます。

## 演習1-19

→別冊テキストP.13

### 解答-エ

ハッシュ法は、データを格納する際、ハッシュ関数によって特定の範囲の値にキーを変換します。そのためデータ1個当たりの探索時間は、表中のデータの個数に関係なく、一定となります。

## 演習1-20

→別冊テキストP.13

### 解答-ア

ハッシュ法とは、ハッシュ関数によりレコードのキー値から格納位置を求める方法です。一連のレコードには同じハッシュ関数を用いるため、格納位置が重複してしまうシノニムが発生する場合があります。しかし、問題にあるようにシノニムの発生を最小とすると、各レコードの格納位置を一意に特定することができます。したがって、アの図が適切です。



## Chapter 3 コンピュータ構成要素

### 演習 1-21

→別冊テキストP.17

#### 解答-ア

キャッシュメモリへの書き込み方式には、ライトスルー方式とライトバック方式があります。ライトスルー方式は、キャッシュメモリと主記憶の内容を同時に書き換える方式です。ライトバック方式は、先にキャッシュメモリだけを書き換えておき、ブロックの入替え時に主記憶を書き換える方式です。

### 演習 1-22

→別冊テキストP.17

#### 解答-イ

ライトスルーは、CPUが主記憶装置にデータを書き込む際に、同時にキャッシュメモリへも同じ内容を書き込む方式です。キャッシュメモリと主記憶装置との間にデータの整合性が常に保たれ、制御も容易です。ライトバックに比べて、書き込み時間は主記憶装置のアクセス時間と同じなので、高速化はあまり望めませんが、読み出しに関しては高速化されます。

### 演習 1-23

→別冊テキストP.17

#### 解答-エ

ライトバック方式は、書き込み命令が実行された時にキャッシュメモリだけを書換えておき、主記憶の書換えはブロックの入れ替え時に行う方式です。そのため、データの読み込みと共に書き込みも高速化されます。ただし、キャッシュメモリと主記憶との間の整合性が保たれず、制御は難しいです。

### 演習 1-24

→別冊テキストP.19

#### 解答-ウ

主記憶の複数のブロックと、キャッシュメモリの複数のブロックを、ランダムに対応付ける方式をフルアソシアティブ方式と呼びます。なお、選択肢イは、ダイレクトマッピング方式に関する記述です。

### 演習 1-25

→別冊テキストP.20

#### 解答-イ

トラックは、ディスク上にデータを記録するための領域で、通常、各ディスクの両面に同心円状に複数存在します。このトラックをいくつかに分割した単位がセクタであり、ディスクにおける物理的な最小単位です。なおシリンダは、一度のシーク動作で同時にアクセス可能な、同心円状のトラックの集合体のことです。

## 演習1-26

→別冊テキストP.20

### 解答-イ

- ・1トラック当たりの記憶容量を算出します。  
 $512\text{バイト}/\text{セクタ} \times 9\text{セクタ}/\text{トラック} = 4,608\text{バイト}/\text{トラック}$
- ・1面当たりの記憶容量を算出します。  
 $4,608\text{バイト}/\text{トラック} \times 160\text{トラック} = 737,280\text{バイト} \approx 0.7\text{Mバイト}$

## 演習1-27

→別冊テキストP.20

### 解答-ウ

レコード50万件分を、50レコードを1ブロックとして記録するため、ブロック数は、  
 $500,000\text{件} \div 50\text{レコード} = 10,000\text{ブロック}$   
この10,000ブロックを記録するのに必要なシリンダ数を求めます。

1ブロック当たりのバイト数

$$50\text{レコード}/\text{ブロック} \times 500\text{バイト}/\text{レコード} = 25,000\text{バイト}/\text{ブロック}$$

1ブロック格納するのに必要なセクタ数（小数以下切上げ）

$$25,000\text{バイト}/\text{ブロック} \div 512\text{バイト}/\text{セクタ} \approx 49\text{セクタ}/\text{ブロック}$$

10,000ブロックを記録するのに必要なセクタ数

$$49\text{セクタ}/\text{ブロック} \times 10,000\text{ブロック} = 490,000\text{セクタ}$$

10,000ブロックを記録するのに必要なトラック数

$$490,000\text{セクタ} \div 25\text{セクタ}/\text{トラック} = 19,600\text{トラック}$$

10,000ブロックを記録するのに必要なシリンダ数

$$19,600\text{トラック} \div 20\text{トラック}/\text{シリンダ} = 980\text{シリンダ}$$

## 演習1-28

→別冊テキストP.20

### 解答-エ

1ブロック当たりのバイト数

$$200\text{バイト} \times 20 + 500\text{バイト} = 500\text{バイト}/\text{ブロック}$$

1トラック当たりのブロック数

$$25200 \div 45 = 5.6 \dots 5\text{ブロック (切捨て)}$$

格納データのブロック数

$$10000 \div 20 = 500\text{件}$$

必要トラック数

$$500\text{件} \div 5\text{ブロック}/\text{トラック} = 100$$

## 演習1-29

→別冊テキストP.20

### 解答-ウ

磁気ディスクを28ビットで表すセクタ番号で管理することになるため、管理可能な最大のセクタ数は、 $2^{28}$ セクタとなります。また、1セクタの容量は512バイトであるため、管理可能な最大の容量は、 $512 \times 2^{28}$ バイトとなります。

ここで、 $512 \times 2^{21}$ を1Gバイトとすると記述されているため、

$$\begin{aligned} & 512 \times 2^{28} \text{バイト} \\ &= 2^7 \times 512 \times 2^{21} \text{バイト} \\ &= 2^7 \text{Gバイト} \\ &= 128 \text{Gバイト} \end{aligned}$$

## 演習1-30

→別冊テキストP.20

### 解答-エ

ハードディスクやフロッピーディスクなどの記憶媒体は、セクタを記録する最小単位ですが、OSが媒体を管理する場合は、セクタ単位では小さすぎるため、複数のセクタをまとめたブロック(この単位をクラスタと呼ぶ)単位で管理します。

本問の場合は、OSの管理単位を1ブロック8セクタ(1ブロックは4,000バイト：500バイト/セクタ×8セクタ/ブロック)と定義しているので、4,000バイト以下のデータを記録する場合は、8セクタ分の領域が必要です。これを計算式で表すと、次のようになります。

$$2,000 \text{バイト} \div 4,000 \text{バイト/ブロック} = 0.5 \cdots 1 \text{ブロック (切り上げ)}$$

$$9,000 \text{バイト} \div 4,000 \text{バイト/ブロック} = 2.25 \cdots 3 \text{ブロック (切り上げ)}$$

となり、割り当てられるセクタ数は、32セクタ(4ブロック×8セクタ/ブロック)となります。

## 解答-イ

この磁気ディスク装置では、16kバイト単位でファイルを管理するため、実データのサイズが16kバイトの倍数でない場合、未使用部分が生じます。

	実データ	必要なデータ領域の大きさ	未使用部分
	22kバイト	32kバイト	10kバイト
	8kバイト	16kバイト	8kバイト
合計	30kバイト	48kバイト	18kバイト

2つのファイルをまとめた48kバイトを全データ領域として考えると、実データの占める割合は次のようになります。

実際の大きさ÷扱う単位での大きさ=使用率

$$30\text{kバイト} \div 48\text{kバイト} = 0.625 \approx 0.63$$

# 演習1-32

## 解答-ウ

各ソフトウェアとデータを格納するための、磁気ディスクの容量を計算します。

$$390(\text{OS}) + 250(\text{ソフトウェア1}) + 350(\text{ソフトウェア2}) + 10(\text{データ}) \\ = 1,000\text{Mバイト}$$

また、2つのソフトウェアを同時に使うことはないので、実行時に必要な作業領域は、大きいほうに合わせて確保します。

$$150(\text{ソフトウェア1の作業領域}) > 120(\text{ソフトウェア2の作業領域})$$

その結果、ソフトウェア用は150(ソフトウェア1の作業領域)となります。

なお、作業領域はOS用も含めて計算します。

$$100(\text{OSの作業領域}) + 150(\text{ソフトウェア1の作業領域}) \\ = 250\text{Mバイト}$$

以上から、最低限必要な磁気ディスクの容量は次のように計算されます。

$$1,000(\text{ソフトウェアとデータの格納領域}) + 250(\text{OSとソフトウェアの作業領域}) \\ = 1,250\text{Mバイト}$$

## 演習1-33

→別冊テキストP.20

### 解答-ウ

ファイルのバックアップを取るために、目的のデータ(今回の条件では30Gバイト)を記憶できるだけの容量があるかどうか確認すると、次のようになります。

- ・ CD-R : 650 ~ 700Mバイト
- ・ DVD-R : 片面4.7Gバイト
- ・ HD : 数10G ~ 100Gバイト
- ・ MO : 230M ~ 2.3Gバイト

上記のデータから判断すると、1つの装置(媒体)でデータを記憶できるものは、HDしか該当しません。また、世代管理を行わないので、CD-RやDVD-Rのような追記型の媒体では書き換えができず、過去に記録したものは不要となり、媒体の無駄が生じてしまいます。さらに、迅速に作業を完了するために、最もアクセス時間が速いものを選択すると、HDを選択するのが最も適しているといえます。

## 演習1-34

→別冊テキストP.23

### 解答-ウ

選ばれなかったものについて解説をしておきます。

- アクセス時間は、磁気ディスクの読み書き時間のことで、シーク時間、サーチ時間、データ転送時間の合計で求めます。
- サーチ時間は、磁気ヘッドが目的のデータがあるトラックに移動したのち、目的のデータがヘッドの位置まで回ってくるまでの時間です。
- データ転送時間は、目的のデータが磁気ヘッドの位置まで回ってきたのち、ヘッドでデータを読み書きする時間です。

## 演習1-35

→別冊テキストP.23

### 解答-イ

サーチ時間(回転待ち時間ともいう)とは、磁気ディスクが回転して、目的のセクタ(記憶装置における最小の記録単位)の先頭が磁気ヘッドの下まで移動する時間のことです。

ディスクの回転スピードが速いほどサーチ時間は短くなり、最大値はディスクが1回転する時間であり、平均値はディスクが $1/2$ 回転する時間です。

選ばれなかったものについて解説をしておきます。

- シリンダとは、ディスク装置で同じ半径のトラック(データを記録するための円筒状の領域)を集めたものです。
- データ転送速度とは、磁気ヘッドが目的のデータを読み出す速度です。
- 磁気ヘッドの位置決め速度とは、磁気ヘッドが目的のトラックまで移動する速度のことです。

## 演習1-36

→別冊テキストP.23

### 解答-ウ

平均待ち時間は、磁気ヘッドを目的のデータが入ったトラックに移動させる平均位置決め時間(シークタイム)と、そのトラックから目的のデータの先頭を探す平均回転待ち時間(サーチタイム)の2つを併せた時間のことを指します。

$$\text{平均回転待ち時間} = 60,000 \text{ミリ秒} / \text{分} \div 4,200 \text{回転} / \text{分} \div 2 = 7.14 \dots \approx 7 \text{ミリ秒}$$

したがって、

$$\text{平均待ち時間} = 5 \text{ミリ秒} + 7 \text{ミリ秒} = 12 \text{ミリ秒}$$

## 演習1-37

→別冊テキストP.23

### 解答-ウ

rpm(revolutions per minute)とは、1分当たりの回転数を表す単位なので、1秒当たりの回転数は次のようになります。

$$6,000 \text{回転} \div 60 \text{秒} = 100 \text{回転} / \text{秒}$$

磁気ディスクでは、ディスクが1回転すると、読み書きを行う磁気ヘッドがディスク上を1周することになり、理論的には1回転に要する時間で1トラック分のデータを読み書きすることができます。

$$1 \text{トラックのデータ量} = 1 \text{セクタのデータ量} \times 1 \text{トラック当たりのセクタ数}$$

$$= 500 \text{バイト} \times 200 \text{セクタ} = 100,000 \text{バイト} = 100 \text{kバイト}$$

この磁気ディスクは1秒間に100回転するため、1秒間での読み書き量は次のようになります。

$$100 \text{kバイト} \times 100 \text{回転} = 10,000 \text{kバイト} = 10 \text{Mバイト} / \text{秒}$$

## 演習1-38

→別冊テキストP.23

### 解答-ウ

$$\text{平均シーク時間} = 10 \text{ミリ秒}$$

$$\text{サーチ時間 (1回転にかかる時間)} = 1 \text{分 (60,000ミリ秒)} \div 5,000 \text{回転} = 12 \text{ミリ秒}$$

$$\text{平均サーチ時間 (平均回転待ち時間)} = 12 \text{ミリ秒} / \text{回転} \div 2 = 6 \text{ミリ秒}$$

$$\text{データ転送速度} = 10 \text{Mバイト} / \text{秒} = 10,000 \text{バイト} / \text{ミリ秒}$$

$$\text{データ転送時間} = 500 \text{バイト} \div 10,000 \text{バイト} / \text{ミリ秒} = 0.05 \text{ミリ秒}$$

$$\text{コントローラ (磁気ディスクを制御する装置) の処理時間} = 2.00 \text{ミリ秒}$$

$$\text{平均アクセス時間} = \text{平均シーク時間} + \text{平均サーチ時間} + \text{データ転送時間}$$

$$= 10 \text{ミリ秒} + 6 \text{ミリ秒} + 0.05 \text{ミリ秒} + 2.00 \text{ミリ秒} = 18.05 \text{ミリ秒}$$

## 演習1-39

→別冊テキストP.23

### 解答-ウ

平均シーク時間=20ミリ秒

サーチ時間(1回転にかかる時間)=1分(60,000ミリ秒)÷6,000回転=10ミリ秒

平均サーチ時間=10ミリ秒÷2=5ミリ秒

データ転送速度=1トラック(1回転)のデータ量÷サーチ時間

=20kバイト÷10ミリ秒=2kバイト/ミリ秒

データ転送時間=4kバイト÷2kバイト/ミリ秒=2ミリ秒

平均アクセス時間=平均シーク時間+平均サーチ時間+データ転送時間

=20ミリ秒+5ミリ秒+2ミリ秒=27ミリ秒

## 演習1-40

→別冊テキストP.23

### 解答-ウ

平均シーク時間=25ミリ秒

サーチ時間(1回転にかかる時間)=1分(60,000ミリ秒)÷2,500回転=24ミリ秒

平均サーチ時間=24ミリ秒÷2=12ミリ秒

データ転送速度=1トラック(1回転)のデータ量÷サーチ時間

=20,000バイト÷24ミリ秒≈833バイト/ミリ秒

データ転送時間=5,000バイト÷833バイト/ミリ秒≈6ミリ秒

平均アクセス時間=平均シーク時間+平均サーチ時間+データ転送時間

=25ミリ秒+12ミリ秒+6ミリ秒=43ミリ秒

## 解答-ア

それぞれの平均アクセス時間を求め、最も短いものを選びます。

- A 平均シーク時間(平均位置決め時間):8ミリ秒  
平均サーチ時間: $60,000\text{ミリ秒} \div 5,000\text{回転} \div 2 = 6\text{ミリ秒}$   
データ転送時間: $50\text{kバイト} \div 10\text{Mバイト/秒} = 0.005\text{秒} = 5\text{ミリ秒}$   
平均アクセス時間: $8\text{ミリ秒} + 6\text{ミリ秒} + 5\text{ミリ秒} = 19\text{ミリ秒}$
- B 平均シーク時間(平均位置決め時間):10ミリ秒  
平均サーチ時間: $60,000\text{ミリ秒} \div 5,000\text{回転} \div 2 = 6\text{ミリ秒}$   
データ転送時間: $50\text{kバイト} \div 5\text{Mバイト/秒} = 0.01\text{秒} = 10\text{ミリ秒}$   
平均アクセス時間: $10\text{ミリ秒} + 6\text{ミリ秒} + 10\text{ミリ秒} = 26\text{ミリ秒}$
- C 平均シーク時間(平均位置決め時間):8ミリ秒  
平均サーチ時間: $60,000\text{ミリ秒} \div 10,000\text{回転} \div 2 = 3\text{ミリ秒}$   
データ転送時間: $50\text{kバイト} \div 5\text{Mバイト/秒} = 0.01\text{秒} = 10\text{ミリ秒}$   
平均アクセス時間: $8\text{ミリ秒} + 3\text{ミリ秒} + 10\text{ミリ秒} = 21\text{ミリ秒}$
- D 平均シーク時間(平均位置決め時間):10ミリ秒  
平均サーチ時間: $60,000\text{ミリ秒} \div 10,000\text{回転} \div 2 = 3\text{ミリ秒}$   
データ転送時間: $50\text{kバイト} \div 5\text{Mバイト/秒} = 0.01\text{秒} = 10\text{ミリ秒}$   
平均アクセス時間: $10\text{ミリ秒} + 3\text{ミリ秒} + 10\text{ミリ秒} = 23\text{ミリ秒}$