

社会と情報
及び
情報の科学

1 次の説明文の空欄 **ア** ～ **コ** に入る最も適切な言葉を選択肢から選び、それぞれ番号で答えなさい。

一般的なコンピュータを構成する装置を機能ごとに制御装置、**ア**、主記憶装置、入力装置、出力装置、外部記憶装置などに大まかに分けることができる。広く普及している PC もその構造は同じで **イ** は入力装置で、**ウ** は出力装置であるが、その両方の機能を持ち合わせるような機能を持つ **エ** は入力装置でもあり出力装置でもある。メインメモリは主記憶装置である。**オ** は制御装置と **ア** の機能が一体化している。しかし最近の PC はグラフィック機能に関しては処理の分岐は少ないが多数のデータを扱うので演算の規模が大きくすべての演算を **オ** では行わず **カ** と呼ばれる専用の部品に代行させてグラフィック機能の処理は行われている。多数のデータを使った規模の大きい演算を実行する能力が注目されて、近年ではグラフィック用の演算だけでなく汎用的な演算ができるように **カ** の機能が進化した。そのため多数のデータを使った規模の大きい演算の需要が大きい AI 分野を中心に **カ** をグラフィック機能以外で活用するという取り組みが広がっている。

スマートフォンもその構造は基本的に同じで前述したように **エ** によって入力装置と出力装置を構成しているが **キ** も搭載しており画像を入力することもできる。また電話の機能を持つため当然ながら音声を入力する **ク** と音声を出力する **ケ** も搭載しているため音声の入出力をさせることもできる。近年のスマートフォンは **カ** の能力が向上しているためそれらの入力と **カ** の能力を活用した、AI 分野の高機能な **コ** が登場しつつある。

選択肢

- | | | | |
|----------|--------|----------|-------|
| ① タッチパネル | ④ マウス | ⑦ ディスプレイ | ⑩ マイク |
| ② スピーカー | ⑤ アプリ | ⑧ CPU | ⑪ GPU |
| ③ カメラ | ⑨ 演算装置 | | |

2

二分探索木に関する説明文をまとめた下記の文章を読み、次の各問い（問1～問4）に答えなさい。

二分探索木は、二分木の各ノードにデータを持たせることにより探索を行えるようにした木である。各ノードが持つデータは、そのデータから出る左部分木にあるどのデータよりも大きく、右部分木のどのデータよりも小さいという条件がありこれを利用して効率的に探索することが可能となっている。

ここで簡単な二分木を作ってみることにする。例えば、1, 3, 5, 7, 8, 10, 14, 16, 18という数があったとする。二分木は

1. 新しいノードの値が現在のノードよりも小さい場合、左の子ノードとして追加
 2. 新しいノードの値が現在のノードよりも大きい場合、右の子ノードとして追加
- という単純なメカニズムで作ることができる。

最初は単純にランダムに「8」を選んだとする。「8」の次に「14」を選んだ場合、「8」よりも「14」の方が大きいため、「8」の右の子ノードとして追加される。その状態を図1に示す。

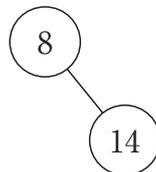


図1 二分木の作り方1

その次に「3」が選ばれたとする。「3」は「8」よりも小さいノードであるため、「8」の左ノードとして追加する。その状態を図2に示す。

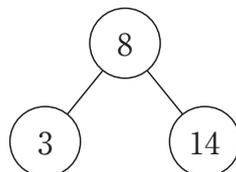


図2 二分木の作り方2

この操作を繰り返せば、図3のような二分木を構築することが可能である。

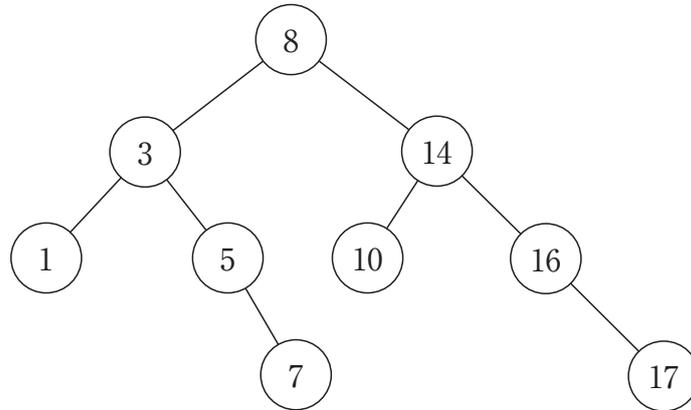


図3 二分木の作り方3

次に、二分探索木の探索方法について示す。

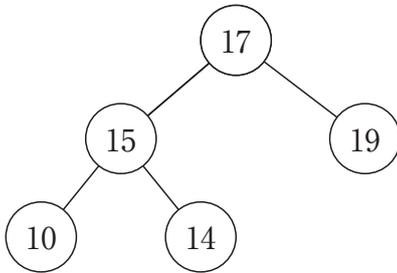
1. 根（一番上のノード）からスタート
2. 探索したい値と現在のノード（着目ノード）を比較
 - (ア) 探索したい値と現在のノードが等しい場合、そのノードを返す
 - (イ) 探索したい値が現在のノードよりも小さい場合、左の子ノードに移動し2.を行う
 - (ウ) 探索したい値が現在のノードよりも大きい場合、右の子ノードに移動し2.を行う

これらについて、値を見つけるかノードがなくなるまで続ける。

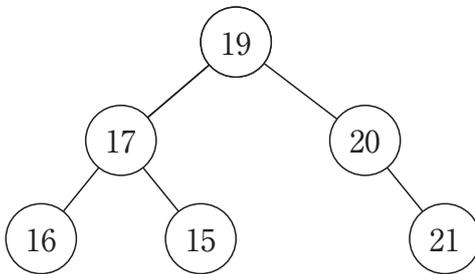
問1 下記①～④のうち、二分探索木になっている二分木はどれか選び、番号をマーク
しなさい。

選択肢

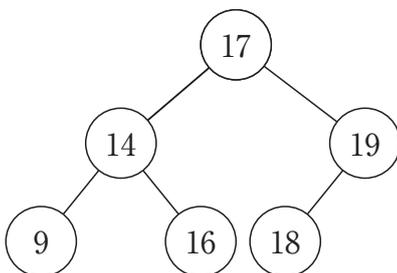
①



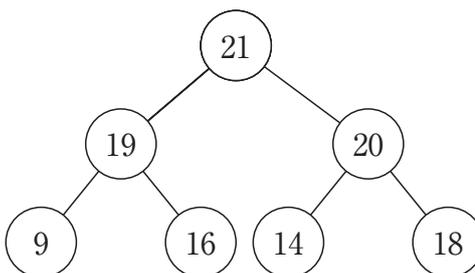
②



③



④



問2 10個のノードからなる下記の図4の二分木の各ノードに1から10までの値を一意に対応するように割り振ったとき、ノード a, b の値の組み合わせとして最も適切なものを選び、番号をマークしなさい。

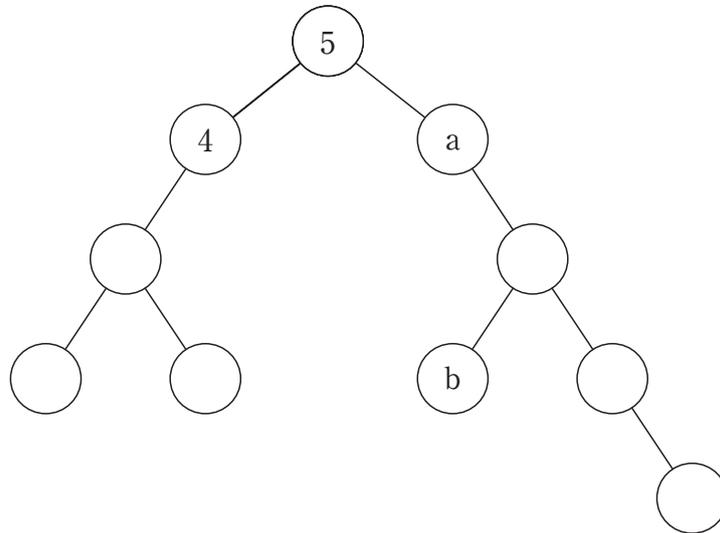


図4 1～10を配置した二分木

選択肢

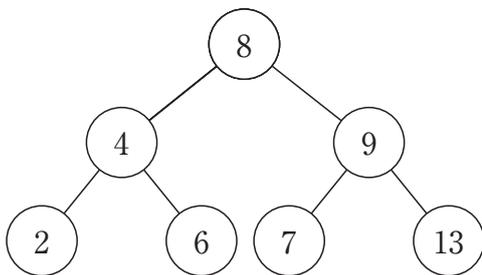
- ① a=6, b=7
- ② a=6, b=8
- ③ a=6, b=9
- ④ a=6, b=10
- ⑤ a=7, b=6
- ⑥ a=7, b=8
- ⑦ a=7, b=9

問3 図3の二分木について7を探索する際、何回のノードの移動が必要かその回数を数字で答えなさい。

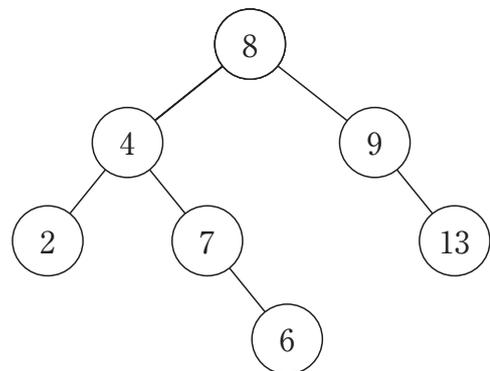
問4 空の二分木に「8」、「13」、「4」、「2」、「9」、「7」、「6」の順にデータを与えたときにできる二分探索木はどれか選び、番号をマークしなさい。

選択肢

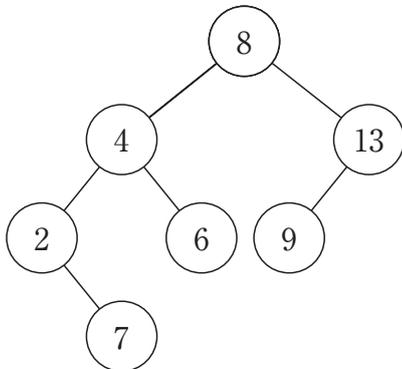
①



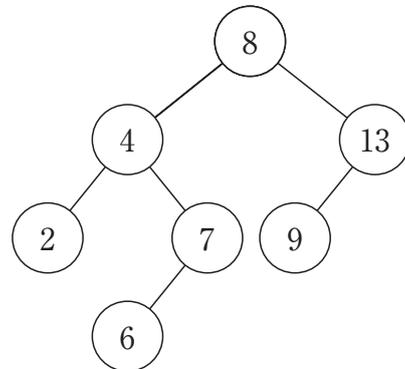
②



③



④



3

近年流行している ChatGPT の仕組みを知りたく、鈴木さんは、GPT は、Generative Pretrained Transformer の略であり、特に、その Transformer と呼ばれる部分の Attention という機構が非常に有効に働いているということを書籍で読んだ。それをまとめた下記の文章を読み、次の各問い（問 1～問 4）に答えなさい。

鈴木さんはインターネットを検索した結果、文献 [1] の Attention is all you need という論文に Attention と呼ばれる機構がどのように計算されるのかが記述してあることがわかった。

Attention は注意機構と呼ばれ、重要な部分を取り出すための計算手法で、次のような式で表されると記述があった。

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Q は queries, K は keys, V は values とあり, d_k は K の次元数と記述があった。

[1] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

しかし、鈴木さんは全く理解ができなかったため、その分野に詳しい先生にざっくりとしたイメージを持つための解説を受けた。

Attention の中でも Q , K , V を全く同じものに置いて計算する Self Attention (自己注意) を理解すると、イメージしやすいと教わった。

まず、この Q , K , V は行列と呼ばれ、列ベクトルと集めたものと考えればよい。列ベクトルを集めたものである行列を作る意味としては、言葉を扱うためには、コンピュータの中で数値に変換する必要があることと教わった。例えば、ある単語と決められた次元数の列ベクトルで表現するのだが、その単語同士の意味が近ければ、変換した列ベクトル同士が近くなるように学習をして割り当てることを Embedding と呼ぶ。

今回は、Self Attention (自己注意) を理解するために、「私の好きな食べ物はカレーです」という文章の中で、それぞれの単語がどのように関係しているか計算するために、上記 Attention (Q, K, V) の式の $\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$ を計算して、この計算の効果を実感することとした。

まずは、「私の好きな食べ物はカレーです」を「私」「の」「好きな」「食べ物」「は」「カレー」「です」という 7 つの単語に分解をした。これらの単語の Embedding する必要があるのだが、本来はこれについても学習をして、適切な列ベクトルに変換すべきである。しかしながら、今回は簡単化のために、Self Attention の結果がうまく出るよう、事前に計算をした結果、次のように各単語を 2 次元の列ベクトルであるとした。

$$\vec{\text{私}} = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \vec{\text{の}} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}, \vec{\text{好きな}} = \begin{bmatrix} 2 \\ 4 \end{bmatrix}, \vec{\text{食べ物}} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \vec{\text{は}} = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \vec{\text{カレー}} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \vec{\text{です}} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Attention の式の中の Q, K, V は行列と呼ばれ、列ベクトルと集めたものであることと、Self Attention (自己注意) を計算する場合、 Q, K, V は全く同じ行列となることから、上記の列ベクトルを集めた行列、 $Q = K = V = \begin{bmatrix} 3 & 1 & 2 & 4 & 3 & 2 & 2 \\ 1 & 4 & 4 & 2 & 3 & 1 & 2 \end{bmatrix}$ ということになる。

Attention (Q, K, V) の式の中にある QK^T の式だが (K^T の T は転置を表すが、この意味を分からなくても下記の説明で解ける)、下記のような図1を作ることと同義となる。

	私	の	好きな	食べ物	は	カレー	です
私	10	7	10	ア	12	7	9
の	7	17	18	12	15	6	10
好きな	10	18	イ	16	18	8	12
食べ物	14	12	ウ	20	18	エ	12
は	12	15	18	18	18	9	12
カレー	7	6	8	10	9	5	6
です	8	10	12	12	12	6	オ

図1 QK^T の計算結果

この具体的な計算方法だが、(1)式としては以下のようなベクトル同士の内積と同値となる。

$$\vec{v}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \vec{v}_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$\vec{v}_1 \cdot \vec{v}_2 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \cdot \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = x_1 \times x_2 + y_1 \times y_2$$

例えば行が「私」、列が「好きな」の数値を計算する場合、答えは10となっているが計算方法は以下のようなになる。

$$\vec{\text{私}} \cdot \vec{\text{好きな}} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 4 \end{bmatrix} = 3 \times 2 + 1 \times 4 = 10$$

各ベクトル同士で上記の計算を行い、図1が完成することが結果的に、 QK^T を計算したことになる。

さらに、プログラミングを行うことによって、⁽²⁾上記の値に $\frac{1}{\sqrt{d_k}}$ をかけた上で *softmax* を計算した。これによって、 $\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$ を計算したことになる。この計算結果をヒートマップで示したものを図2に示す。

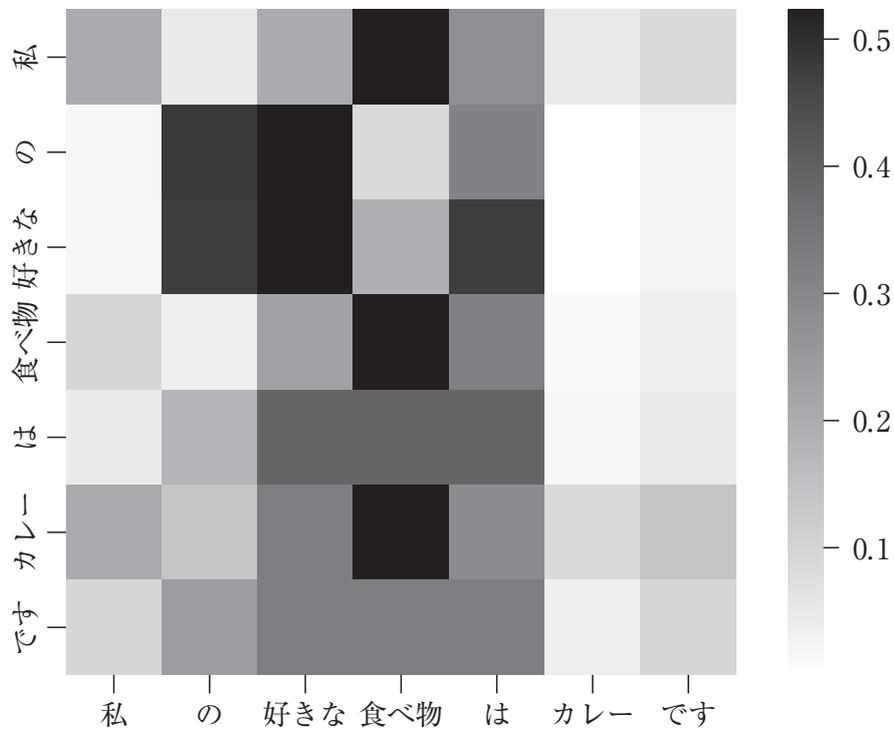


図2 $\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$ の計算結果のヒートマップ

⁽³⁾図2の結果から、「私の好きな食べ物はカレーです」という文章の中の単語同士の重要な部分が導出できていることが分かる。

問1 図1の ~ について、適切な数値を下記の選択肢から選び、それぞれ番号をマークしなさい（同じ数値を選択しても構わない）。

選択肢

- | | | | | |
|------|------|------|------|------|
| ① 8 | ② 9 | ③ 10 | ④ 12 | ⑤ 13 |
| ⑥ 14 | ⑦ 15 | ⑧ 16 | ⑨ 18 | ⑩ 20 |

問2 下線部(1)について、このようにそれぞれのベクトル同士の内積の計算をすることは何を意味しているのか最も正しいものを下記の選択肢から選び、番号をマークしなさい。

選択肢

- ① 単語の分割した分割の方法の妥当さを表現しようとしている
- ② 分割した単語を Embedding した値の妥当性を表現しようとしている
- ③ 分割した単語同士の近さ、遠さを数値で表現しようとしている
- ④ 分割した単語に付与された重みを総量として同様の値に調整しようとしている

問3 下線部(2)について、このような計算をすることは何を意味しているのか最も正しいものを下記の選択肢から選び、番号をマークしなさい。

選択肢

- ① 単語の分割した分割の方法の妥当さを表現しようとしている
- ② 分割した単語を Embedding した値の妥当性を表現しようとしている
- ③ 分割した単語同士の近さ、遠さを数値で表現しようとしている
- ④ 分割した単語に付与された重みを総量として同様の値に調整しようとしている

問4 下線部(3)について、図2のヒートマップから何が導出され、文章生成にいかされているかの考察について正しく述べられていないものを下記の選択肢から選び、番号をマークしなさい。

選択肢

- ① 「カレー」行で一番大きい値が「食べ物」であるため、「カレー」と「食べ物」が近いという単語の意味的なものを抽出できた
- ② 「カレー」行で小さな値が「の」であるため、「カレー」と「の」が遠いという単語の意味的なものを抽出できた
- ③ この計算は、周辺の単語しか計算ができず、長い文章に含まれる単語同士の関係性を抽出することができない
- ④ 「私」の行の「私」、「の」の行の「の」など同じ単語同士は対角上に配置され、比較的大きな値を取ることが多い

4

コンピュータでの画像の基礎的な描画のプログラミングに関する以下の文章を読み、次の各問い（問1～問4）に答えなさい。

コンピュータの画面について x 座標と y 座標を用いた x y 平面を用いて画面上の位置を定義する。コンピュータの画面の左上の角を原点 $(0, 0)$ とし、左下の角を $(0, 100)$ 、右上の角を $(100, 0)$ 、右下の角を $(100, 100)$ とする。ただしコンピュータの画面の縦と横の比は $1 : 1$ の正方形の画面とする。

コンピュータの画面に対する描画に使える命令は以下とする。

- ・座標 (a, b) から座標 (c, d) に黒い直線を引く

コンピュータの画面に対する描画のプログラムは以下のように記述するものとする。

例1：画面の4つの角について対角同士を直線でつないだ \times を描画する場合

- ・座標 $(0, 0)$ から座標 $(100, 100)$ に黒い直線を引く
- ・座標 $(0, 100)$ から座標 $(100, 0)$ に黒い直線を引く

例2：画面の1辺の大きさの半分の大きさの辺をもつ正方形を画面の中心に描画する場合

- ・座標 $(25, 25)$ から座標 $(25, 75)$ に黒い直線を引く
- ・座標 $(25, 75)$ から座標 $(75, 75)$ に黒い直線を引く
- ・座標 $(75, 75)$ から座標 $(75, 25)$ に黒い直線を引く
- ・座標 $(75, 25)$ から座標 $(25, 25)$ に黒い直線を引く

問1 次の文章を読み、空欄に入る最も適切な語句を選択肢の中から選び、その番号をマークしなさい。, ,

一般に、写真などの画像は、平面空間上で色の濃淡が連続的に分布したアナログデータである。画像はまず画像を画素という小さな区画に分割し、それぞれの濃淡をよみとる を行い、次に区画の濃淡を数値に変換する をする。最後に した数値を2進法に変換する を行うことでデジタルデータに変換することができる。

選択肢

- ① 量子化 ② 符号化 ③ 暗号化 ④ 標本化

描画プログラム A を以下のように定義する。

- ・座標 (25, 75) から座標 (75, 75) に黒い直線を引く
- ・座標 (25, 75) から座標 (25, 25) に黒い直線を引く
- ・座標 (75, 75) から座標 (75, 25) に黒い直線を引く

問2 描画プログラム A は口の字型のうち一辺が閉じていないコの字型の図形が表示されるが閉じていない辺として正しいものを選択肢の中から選び、その番号をマークしなさい。

- ① 口の字型の上の辺
- ② 口の字型の下の辺
- ③ 口の字型の左の辺
- ④ 口の字型の右の辺

描画プログラム B を以下のように定義する。

- ・座標 (25, 75) から座標 (75, 75) に黒い直線を引く
- ・座標 (75, 25) から座標 (25, 25) に黒い直線を引く
- ・座標 (25, 75) から座標 (75, 25) に黒い直線を引く

問3 描画プログラム B はアルファベットに近い図形を表示する。文字の回転や反転をしていないとして、選択肢の中からもっとも近い形状の文字を選び、その番号をマークしなさい。

- ① N ② S ③ Z ④ Y

問4 90度回転したり上下反転したりしていない正しい向きのアルファベットの「M」に近い図形を描画する描画プログラムとしてもっとも適切なものを選択肢の中から選び、その番号をマークしなさい。 カ

①

- ・座標 (25, 75) から座標 (25, 25) に黒い直線を引く
- ・座標 (75, 75) から座標 (75, 25) に黒い直線を引く
- ・座標 (25, 75) から座標 (50, 25) に黒い直線を引く
- ・座標 (75, 75) から座標 (50, 25) に黒い直線を引く

②

- ・座標 (25, 75) から座標 (25, 25) に黒い直線を引く
- ・座標 (75, 75) から座標 (75, 25) に黒い直線を引く
- ・座標 (50, 75) から座標 (25, 25) に黒い直線を引く
- ・座標 (50, 75) から座標 (75, 25) に黒い直線を引く

③

- ・座標 (75, 25) から座標 (25, 25) に黒い直線を引く
- ・座標 (75, 75) から座標 (25, 75) に黒い直線を引く
- ・座標 (75, 50) から座標 (25, 25) に黒い直線を引く
- ・座標 (75, 50) から座標 (25, 75) に黒い直線を引く

④

- ・座標 (75, 25) から座標 (25, 25) に黒い直線を引く
- ・座標 (75, 75) から座標 (25, 75) に黒い直線を引く
- ・座標 (75, 25) から座標 (25, 50) に黒い直線を引く
- ・座標 (75, 75) から座標 (25, 50) に黒い直線を引く