

広島市立大学大学院情報科学研究科
博士前期課程 一般入試

口頭試問 サンプル問題B

【2025 年 6 月 30 日公表】

本問題は、広島市立大学大学院情報科学研究科博士前期課程一般入試の第1回募集と第2回募集で実施する口頭試問のサンプル問題です。これらの入学試験を受験することを考えている方は参考にしてください。

1. このサンプル問題は、この表紙を除き3ページあります。
2. このサンプル問題は、A群【数学】とB群【データ構造とアルゴリズム】の2つの群で構成しており、各群2問ずつあります。
3. 入試区分により選択・解答する問題数が異なります。詳しくは、各入試区分の学生募集要項をご覧ください。

選択問題 A【数学】

次の問 A1, 問 A2 のいずれか 1 つを選んで解答せよ.

問 A1

次の連立 1 次方程式について, 以下の問いに答えよ.

$$\begin{cases} x + 2y + 3z = 6 \\ 2x + 5y + 8z = 17 \\ 3x + 7y + 11z = 23 \end{cases}$$

- (1) この連立 1 次方程式の係数行列と拡大係数行列について, それぞれの階数 (rank) を求めよ.
- (2) この連立 1 次方程式が解をもつかどうかを判定せよ. また, 解をもつ場合は一般解を求めよ.

問 A2

関数 $f(x) = (1+x)^\alpha$ ($x > -1$, α は実数) について, 以下の問いに答えよ.

- (1) $f(x)$ を 1, 2, 3 回微分した関数を, それぞれ $f'(x)$, $f''(x)$, $f'''(x)$ で表す.
このとき, $f(0)$, $f'(0)$, $f''(0)$, $f'''(0)$ を求めよ.
- (2) $-1 < x < 1$ のとき, $f(x)$ をマクローリン展開 (Maclaurin expansion) せよ.
ただし, 0 でない最初の 4 項を明記すること.

選択問題 B【データ構造とアルゴリズム】

次の問 B1、問 B2 のいずれか 1 つを選んで解答せよ。

問 B1

サイズ n の配列で n 個のデータを格納することを考える。一般に、配列内のデータの並びによって、この配列に対する操作（データの「検索」「追加」「削除」など）に要する計算の手間は異なる。ここでは、

- (a) 配列内のデータが整列されている状態を維持する
- (b) 並びに関する条件がない（整列済みとは限らない）

の 2 通りの場合を想定する。

このことを踏まえて次の問いに答えよ。

- (1) 配列の中から指定したデータ val の存在位置を求める「検索」のアルゴリズムを、(a)、(b) の 2 つの場合について説明せよ。
- (2) (1) で示した 2 つのアルゴリズムの最悪時の時間計算量を、オーダ記法を用いてそれぞれ示せ。
- (3) 上述の「検索」に加えて、以下の 2 つの操作のアルゴリズムを踏まえながら、2 つの場合 (a)、(b) における、時間計算量に関するトレードオフについて議論せよ。

追加：データ val を配列に加える。配列のサイズは操作後に 1 つ増加する。

削除：データ val を配列から取り出す。配列のサイズは操作後に 1 つ減少する。

問 B2

図 1 は、行 1 から行 3 にある 3 つの配列 `element`, `left`, `right` で表現される 2 分木 T と、それに対して何らかの操作を行う関数 `traverse()` を示している。配列の添え字は 0 から始まるものとする。

各配列の添え字 i の値 `element[i]`, `left[i]`, `right[i]` はそれぞれ、節点 i に格納されている要素、節点 i の左の子、

節点 i の右の子を表している。例えば、`element[3]` には D が格納されており、その左の子 `left[3]` はなく (-1)、右の子 `right[3]` は節点 4 であり、その節点が格納する要素は E である。

このとき以下の問いに答えよ。

- (1) この 2 分木 T を図示せよ。
- (2) この関数 `traverse(node)` は 2 分木 T に対してどのような操作を行うか、引数 `node` を使って説明せよ。
- (3) この関数の 7 行目と 8 行目の間に `print(element[node])` を加えた上で、`traverse(0)` を実行する。このときに得られる出力を示せ。

```

1  element = ['A','B','C','D','E','F','G']
2  left = [1,2,-1,-1,5,-1,-1]
3  right = [-1,3,-1,4,6,-1,-1]
4
5  def traverse(node):
6      if(node == -1): return
7      traverse(right[node])
8      traverse(left[node])

```

図 1: プログラム