情報科学

情報系院試　ロードマップシリーズ

6か月ロードマップと類題100問

大阪大学大学院 情報科学研究科（数学・専門科目）

ロードマップ

第1章　試験の構成と出題マップ
第2章　1月から7月までの計画
第3章　数学（線形代数・微分積分・確率）
第4章　専門科目（情報工学）
第5章　仕上げの模擬セット

2027年 夏入試向け

出題マップ・月別計画・類題100問

院試 hub

利用上の注意

本書は院試 hub が独自に作成した基礎ノートと練習問題です。大学・研究科等の公式資料ではありません。各大学の過去問の本文・原図は収録していません。出題分野の頻度は、院試 hub が解答を公開している実施回の分類に基づきます。本書の無断転載、再配布、共有を禁じます。

目次

出題分野の頻度 6

第 1 章　試験の構成と出題マップ 7

専攻・試験科目・試験時間 7

分野ごとの頻度と本書の配分 8

年度 × 大問の出題分野 9

第 2 章　1 月から 7 月までの計画 10

計画の前提と進め方 10

第 3 章　数学（線形代数・微分積分・確率） 12

線形代数：階数・固有値・二次形式 12

問題 1　パラメータを含む連立一次方程式 13

問題 2　核と像の基底と次元定理 14

問題 3　3 次正方行列の対角化 15

問題 4　重根をもつ行列の対角化可能性 16

問題 5　推移行列の冪と極限 17

問題 6　ケーリー・ハミルトンの定理による冪の計算 19

問題 7　対角化できない行列の冪と級数 20

問題 8　行列の指数関数と連立微分方程式 21

問題 9　多項式空間上の線形写像の固有値 22

問題 10　重根をもつ実対称行列の直交対角化 23

出題分野の頻度

院試 hub が解答を公開している実施回の大問を分類した問数。1 大問を 1 分野に数えている。

分野	問数	出題年度
数学：固有値・対角化・行列の冪	9	2025、2024、2022、2021、2020、2019、2016
数学：線形空間・核と像・階数	7	2025、2022、2021、2018、2017、2016
数学：重積分・体積	7	2024、2023、2022、2020、2019、2017、2016
数学：1 変数の積分・広義積分	7	2023、2021、2020、2019、2018、2017、2016
数学：二変数関数の極値	6	2025、2024、2023、2020、2018、2016
情報工学：アルゴリズムとデータ構造	6	2025、2024、2023、2022、2021、2020
情報工学：離散構造	6	2025、2024、2023、2022、2021、2020
情報工学：オートマトン・形式言語	6	2025、2024、2023、2022、2021、2020
情報工学：ネットワーク	6	2025、2024、2023、2022、2021、2020
情報工学：電子回路と論理設計	6	2025、2024、2023、2022、2021、2020
情報工学：信号処理と変換	6	2025、2024、2023、2022、2021、2020
数学：数列・級数・関数列	4	2025、2024、2023、2017
数学：常微分方程式	4	2025、2022、2021、2018
情報工学：OS（記憶管理・排他制御）	4	2024、2023、2022、2021
数学：二次形式・正定値行列	3	2024、2023、2017
数学：微分可能性・導関数の極限	3	2022、2021、2019
情報工学：計算機アーキテクチャ	2	2025、2020

問題 27　積分順序の交換

阪大 情報基礎数学 2017 第 1 問・2024 第 1 問と同じ分野（重積分）

問題

次の累次積分を求めよ。

(1) $\displaystyle\int_0^1 \left(\int_x^1 e^{y^2}\, dy \right) dx$　　(2) $\displaystyle\int_0^1 \left(\int_{\sqrt{x}}^1 \sqrt{1+y^3}\, dy \right) dx$

解答

(1)　積分領域は $D = \{(x,y) \mid 0 \le x \le 1,\ x \le y \le 1\} = \{(x,y) \mid 0 \le y \le 1,\ 0 \le x \le y\}$ である。被積分関数は D 上で連続なので、フビニの定理により順序を交換できて

$$\int_0^1 \left(\int_0^y e^{y^2} dx \right) dy = \int_0^1 y e^{y^2} dy = \left[\tfrac{1}{2} e^{y^2} \right]_0^1 = \frac{e-1}{2}.$$

(2)　領域は $\{0 \le x \le 1,\ \sqrt{x} \le y \le 1\} = \{0 \le y \le 1,\ 0 \le x \le y^2\}$。順序を交換して

$$\int_0^1 y^2 \sqrt{1+y^3}\, dy = \left[\tfrac{2}{9} (1+y^3)^{3/2} \right]_0^1 = \frac{2}{9} \left(2\sqrt{2} - 1 \right).$$

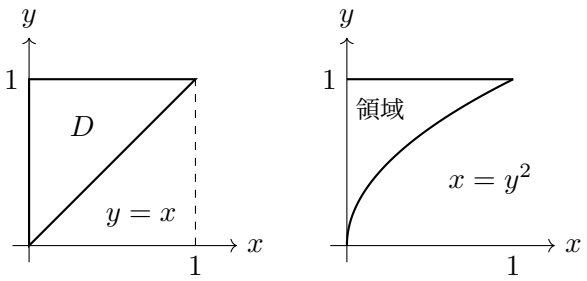
答　(1) $\dfrac{e-1}{2}$。(2) $\dfrac{2(2\sqrt{2}-1)}{9}$。

解説

e^{y^2} と $\sqrt{1+y^3}$ には初等関数の原始関数がないので、与えられた順序では内側の積分が計算できない。順序を交換すると、内側の積分は x についての定数の積分になり、長さ y や y^2 が掛かる。その因子がちょうど置換積分 $u = y^2$、$u = 1+y^3$ の微分に当たる。順序交換の問題の多くは、この組み合わせで作られている。

交換の手順は、領域の境界を「x を固定して y の範囲」から「y を固定して x の範囲」に読み替えることで、図を描くのが確実である。(1) の D は頂点 $(0,0), (0,1), (1,1)$ の三角形で、y を固定すると x は 0 から y まで動く。(2) の領域は曲線 $x = y^2$ の左側で、y を固定すると x は 0 から y^2 まで動く。$\sqrt{x} \le y$ を $x \le y^2$ に直すときに $y \ge 0$ を使っている。

範囲を $0 \le x \le y^2$ ではなく $y^2 \le x \le 1$ と逆に読むと、(2) は $\int_0^1 (1-y^2)\sqrt{1+y^3}\, dy$ となり、原始関数が求まらずに行き詰まる。行き詰まったら領域の読み替えを疑えばよい。(1) の値 0.859 は、$1 \le e^{y^2} \le e$ と三角形の面積 $\frac{1}{2}$ から得られる範囲 $[0.5, 1.36]$ に入っている。



解説

線形探査の弱点は、埋まったセルが連続した塊（一次クラスタ）を作り、塊に当たったキーが塊の端まで進んで塊をさらに伸ばすことにある。この表では添字 0〜5 が一つの塊になり、$h=0$ に写る 33 は 6 回もの探査を要した。

負荷率 $\alpha=\frac{9}{11}\approx 0.82$ のとき、一様ハッシュの近似による成功探索の期待値 $\frac{1}{2}\left(1+\frac{1}{1-\alpha}\right)\approx 3.3$ と比べても、平均 2.22 はこの具体例の値にすぎない。挿入順を変えると平均も変わる。

空セルは、線形探査で「この先にキーはない」という停止の合図に使われている。削除で空セルを作ると、その合図が嘘になる。印つきの削除（墓石）は正しいが、削除が多いと印のセルが増えて探索が長くなるので、ある程度たまったら表全体を作り直す。チェイン法（同じハッシュ値のキーを連結リストでつなぐ方法）では、この問題は起きない。探査列が必ず止まるかどうかは、2023 年度の第 1 問でも論点になっていた。

問題 58　二分探索木の挿入・削除と高さ

阪大 情報工学 2024 第 1 問・情報数理学 2023 第 1 問と同じ分野（木構造・二分探索木）

問題

空の二分探索木に $50, 30, 70, 20, 40, 60, 80, 35, 45, 65$ の順でキーを挿入する。

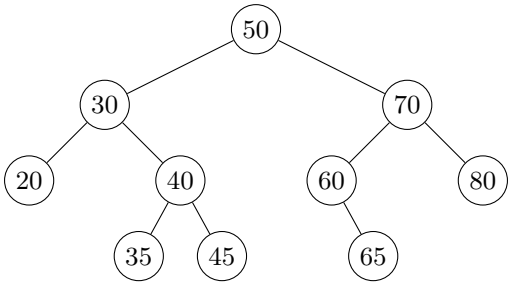
(1) 得られる木を図示し、高さ（根から最も深い葉までの辺の数）と、45 を探索するときの比較回数を求めよ。

(2) キー 30 を削除せよ（子が 2 つある節点は、右部分木の最小のキーで置き換える）。続けて 70 を削除せよ。それぞれの後の木を先行順（根→左→右）で書け。

(3) n 個のキーの挿入順によって、高さがとりうる最小値と最大値を求めよ。

解答

(1)　木は次のとおりで、高さは 3。45 の探索は 50, 30, 40, 45 と比べて 4 回。



(2)　30 の右部分木の最小は 35。30 を 35 で置き換え、元の 35（葉）を消す。先行順は $50, 35, 20, 40, 45, 70, 60, 65, 80$。次に 70 の右部分木の最小は 80（左の子がない）。70 を 80 で置き換え、元の 80 を消すと、先行順は $50, 35, 20, 40, 45, 80, 60, 65$。

(3)　高さ h の二分木の節点数は $2^{h+1}-1$ 以下なので $h\geq\lceil\log_2(n+1)\rceil-1$、この値は均衡した挿入順で達成される。昇順に挿入すると各節点が右の子を 1 つだけもつ鎖になり、高さは最大の $n-1$。

答　(1) 高さ 3、比較 4 回。(2) $50, 35, 20, 40, 45, 70, 60, 65, 80$ の後、$50, 35, 20, 40, 45, 80, 60, 65$。(3) 最小 $\lceil\log_2(n+1)\rceil-1$、最大 $n-1$。

解説

二分探索木の性質は「各節点で、左部分木のキーはすべて小さく、右部分木のキーはすべて大きい」ことで、中間順（左→根→右）にたどると昇順に並ぶ。削除で右部分木の最小（中間順の次の値、後続）を使うのは、それが左部分木のどのキーより大きく、右部分木の残りのどのキーより小さいからで、置き換えても性質が保たれる。後続は左の子をもたないので、元の位置からの削除は子が 1 つ以下の簡単な場合に帰着する。2 回目の削除で、70 の後続 80 は右部分木の根そのもので、左の子がない。このとき「右部分木の最小」を探して左へ降りる手順は 1 歩も進まない。ここで 70 の左の子 60 を後続と答える誤りがある。後続は 70 より大きいキーのうち最小のもので、右部分木の中から選ぶ。左部分木の 60 と 65 はどちらも 70 より小さい。

操作の計算量はどれも高さに比例する。最悪の高さ $n-1$ を避けるには、AVL 木や赤黒木のように回転で高さを $O(\log n)$ に保つ。ランダムな順序で挿入した場合の高さの期待値も $O(\log n)$ である。

問題 59　循環配列のキューと 2 本のスタックによるキュー

阪大 情報工学 2021 第 1 問と同じ分野（キュー）

問題

(1) 大きさ 5 の配列 $q[0..4]$ と、先頭の添字 `head`、要素数 `count` でキューを実装する。追加は $q[(\texttt{head}+\texttt{count}) \bmod 5]$ に書いて `count` を 1 増やし、取り出しは $q[\texttt{head}]$ を返して $\texttt{head} \leftarrow (\texttt{head}+1) \bmod 5$、`count` を 1 減らす。初期状態 $\texttt{head}=0,\ \texttt{count}=0$ から、追加 A, B, C、取り出し 2 回、追加 D, E, F, G を行った後の配列、`head`、`count` を求めよ。続けて追加 H を行うとどうなるか。

(2) スタック $S_{\text{in}}, S_{\text{out}}$ でキューを実装する。追加は S_{in} に積む。取り出しは、S_{out} が空なら S_{in} の要素をすべて 1 個ずつ S_{out} に移してから S_{out} の先頭を取る。n 回の操作（追加と取り出しの任意の列）の、スタック操作（積む・取る）の総数が $4n$ 以下であることを示せ。

解答

(1)　A, B, C を添字 $0, 1, 2$ に置き、2 回の取り出しで A, B を返して $\texttt{head}=2$、$\texttt{count}=1$。D, E, F, G はそれぞれ添字 $(2+1) \bmod 5 = 3$、4、0、1 に書かれ、$\texttt{count}=5$。配列は (F, G, C, D, E)（取り出した A, B の位置は上書き済み）、$\texttt{head}=2$。ここで H を追加しようとすると $\texttt{count}=5$ で満杯なので、追加できない（あふれの処理が要る）。

(2)　各要素に注目する。要素 1 個は、生涯で S_{in} に積まれ（1 回）、S_{in} から取られ（1 回）、S_{out} に積まれ（1 回）、S_{out} から取られる（1 回）、の高々 4 回しかスタック操作を受けない。n 回の操作で追加される要素は高々 n 個なので、総数は $4n$ 以下である。

答　(1) 配列 (F, G, C, D, E)、$\texttt{head}=2$、$\texttt{count}=5$。H は満杯で追加できない。(2) 上の議論により総数 $\le 4n$。

解説

循環配列では、添字を剰余で回すことで、取り出しで空いた先頭側のセルを再利用する。`head` と末尾の添字 `tail` だけで実装すると、満杯と空がどちらも $\texttt{head}=\texttt{tail}$ になって区別できない。要素数 `count` を別に持つか、配列の 1 セルを常に空けておくことで区別する。(1) で `count` を持たせたのはこのためである。