算法数理工学 基本演習 略解

問題 1. $\dfrac{c^n}{n^p} = f(n)$ とおく．$c^n = n^p f(n)$ の両辺の対数をとると $n \log c = p \log n + \log f(n)$ すなわち

$$\log f(n) = n \log c - p \log n$$

である．n の関数 $n \log c$ の傾きは定数 $\log c$ であるのに対し，n の関数 $p \log n$ の傾きは p/n であり，$n \to \infty$ のときこの傾きは 0 に収束する．したがって，上式の右辺はどれだけでも大きくなり得るから $\lim_{n\to\infty} \log f(n) = \infty$ である．ゆえに $\lim_{n\to\infty} f(n) = \infty$ である．

(このことから，次の事実がわかる：大きさ n の問題を解く二つのアルゴリズムがあって，一方の計算時間が c^n に比例し，もう一方が n^p に比例するならば，十分大きな n に対しては，後者のアルゴリズムの方が短い時間で問題を解くことができる．)

問題 2.

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$$

より

$$\begin{aligned}
\lim_{n\to\infty} \frac{n^p}{\log_c n} &= \lim_{n\to\infty} \frac{\dfrac{\mathrm{d}}{\mathrm{d}n}(n^p)}{\dfrac{\mathrm{d}}{\mathrm{d}n}(\log_c n)} = \lim_{n\to\infty} \frac{pn^{p-1}}{\dfrac{1}{\log_e c}\dfrac{1}{n}} \\
&= \lim_{n\to\infty} pn^p \log_e c = \infty.
\end{aligned}$$

(したがって，たとえば計算時間が $n^{0.0001}$ に比例するアルゴリズムよりも，計算時間が $\log_2 n$ に比例するアルゴリズムの方が，十分大きな n に対しては速い．)

問題 3. x 以上の最小の整数を $\lceil x \rceil$ で表す．このとき，

$$\begin{aligned}
n! &\geq n(n-1)(n-2)\cdots\left(\left\lceil \frac{n}{2} \right\rceil\right) \geq \left(\frac{n}{2}\right)^{\frac{n}{2}}, \\
c^n &= (c^2)^{\frac{n}{2}}
\end{aligned}$$

である．だから

$$\frac{n!}{c^n} \geq \left(\frac{n}{2c^2}\right)^{\frac{n}{2}} \to \infty \quad (n \to \infty \ \text{のとき}).$$

(したがって，大きな n に対しては，計算時間が $n!$ に比例するアルゴリズムは，計算時間が n の指数関数に比例するアルゴリズムより遅い．)

問題 4. (1) $(3n^2 + 5n + 6)/n^2 = 3 + 5/n + 6/n^2 \leq 14 < 15$ だから正しい．

(2) $(3n^2 + 5n + 6)/n^3 = 3/n + 5/n^2 + 6/n^3 \leq 14 < 15$ だから正しい．

(3) $(3n^2 + 5n + 6)/(n\sqrt{n}) = 3\sqrt{n} + 5/\sqrt{n} + 6/(n\sqrt{n})$ であるが，第 1 項が $n \to \infty$ のとき発散するから，正しくない．

(4) $(2^n + 10n^3)/n^3 = 2^n/n^3 + 10$ であるが，$n \to \infty$ のとき $2^n/n^3$ は発散するから正しくない．

(5) $(n^2 + n\log n)/(n\log n) = n/\log n + 1$ だが，$n \to \infty$ のとき第一項は発散するから正しくない．

(6) $(n\sqrt{n} + n(\log n)^2)/(n(\log n)^2) = \sqrt{n}/(\log n)^2 + 1$ だが，$n \to \infty$ のとき第一項は発散するから正しくない．

問題 5. $\log_a n = \dfrac{\log_b n}{\log_b a}$ であり，$\dfrac{1}{\log_b a}$ は n によらない定数であるから，$f(n) = \mathrm{O}(\log_b n)$ である．(したがって，n の対数のオーダは，対数の底の選び方に依存しない)

問題 6. (1) $2.3n^3 + 1.6n^2 + 3 = \mathrm{O}(n^3)$.

(2) $3n^2 + 2.6n\sqrt{n}\log n = \mathrm{O}(n^2)$.

(3) $n\log n + \sqrt{n}(\log n)^3 = \mathrm{O}(n\log n)$.

(4) $3n^2 + 2^n = \mathrm{O}(2^n)$.

(5) $\log n = x$ とおくと $3\log n + 2(\log\log n)^3 = 3x + 2(\log x)^3 = \mathrm{O}(x)$ である．したがって，$3\log n + 2(\log\log n)^3 = \mathrm{O}(\log n)$.

問題 7. $f(n)$ の形を

$$f(n) = \alpha^n$$

と仮定する．これを問題の第 2 式に代入すると

$$\alpha^n = \alpha^{n-1} + \alpha^{n-2}$$

より $\alpha^2 - \alpha - 1 = 0$ が得られるから，

$$\alpha = \frac{1 \pm \sqrt{5}}{2}$$

となる．そこで，定数 A, B を用いて，あらためて $f(n)$ を

$$f(n) = A\left(\frac{1+\sqrt{5}}{2}\right)^n + B\left(\frac{1-\sqrt{5}}{2}\right)^n$$

とおき，$f(1) = f(2) = 1$ に代入すると，$A = \dfrac{1}{\sqrt{5}}$, $B = \dfrac{-1}{\sqrt{5}}$ が得られるから

$$f(n) = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$$

となる．

(この数列 $f(n)$, $n = 1, 2, 3, \cdots$, はフィボナッチ数列とよばれる．上の結果から $f(n) = \mathrm{O}\left(\left(\dfrac{1+\sqrt{5}}{2}\right)^n\right)$ であることがわかる．

問題 8. できるだけ低い高さでたくさんの頂点をもつ 2 進木を作るためには，根に近いところから 2 個の子をもつ頂点だけを作っていくとよい．高さ 0 の頂点は根 v_0 のみ，高さ 1 の頂点は v_0 の子 2 個，高さ 2 の頂点はその子 2^2 個，$\cdots$ となるから，求める高さの最小値は，

$$n \leq 1 + 2 + 2^2 + \cdots + 2^k$$

となる最小の k である．この不等式の右辺は $2^{k+1} - 1$ であるから，$n + 1 \leq 2^{k+1}$，すなわち $k \leq \log_2(n+1) - 1$．したがって，高さの最小値は $\lceil \log_2(n+1) \rceil - 1$ である．ただし $\lceil x \rceil$ は x 以上の最小の整数を表す．

(このことから，n 個の頂点をもつ 2 進木は高さ $\mathrm{O}(\log n)$ で作ることができる．)

問題 9. MOVE(n,A,B,C) の時間複雑度を $f(n)$ とおく．$f(1) = c$ (定数) である．$n \geq 2$ に対しては，MOVE(n,A,B,C) の中では，n のかわりに $n-1$ とおいてこの手続きを 2 回呼び出しているから $f(n) = 2f(n-1)$ である．したがって $f(n) = 2^{n-1}f(1) = \mathrm{O}(2^n)$ である．

問題 10. 2 行目では，$1 \leq i < j \leq n$ を満たすすべての対 (i,j) について代入操作が行われる．そのような対は $\mathrm{O}(n^2)$ 個あり，それぞれの代入操作は定数時間で実行できる．したがって，その計算時間は $\mathrm{O}(n^2)$ である．一方，5 行目の操作 (これも定数時間で実行できる) は，$1 \leq k \leq n$, $1 \leq i < j \leq n$ を満たすすべての (i,j,k) の組に対して行われる．このような組は $\mathrm{O}(n^3)$ 個あるから，3 行目から 5 行目までの計算時間は $\mathrm{O}(n^3)$ である．したがって，このアルゴリズムの時間複雑度は $\mathrm{O}(n^3)$ である．

問題 11. 漸化式の n を $\frac{n}{2}, \frac{n}{2^2}, \cdots$ で置き換えた式を利用すると，$f(n)$ は次のように変形できる．

$$\begin{aligned}
f(n) &= 2f\left(\frac{n}{2}\right) + cn \\
&= 2\left(2f\left(\frac{n}{2^2}\right) + \frac{cn}{2}\right) + cn \\
&= 2\left(2\left(2f\left(\frac{n}{2^3}\right) + \frac{cn}{2^2}\right) + \frac{cn}{2}\right) + cn \\
&\;\;\vdots \\
&= 2\left(2\left(\cdots\left(2f\left(\frac{n}{2^k}\right) + \frac{cn}{2^{k-1}}\right) + \cdots\right) + \frac{cn}{2}\right) + cn \\
&= 2^k f\left(\frac{n}{2^k}\right) + cn(1 + 1 + \cdots + 1) \\
&= 2^k f(1) + cnk = cn + cn\log_2 n.
\end{aligned}$$

(したがって，大きさ n の問題が，大きさ $n/2$ の小問題 2 個に分解でき，その小問題の解を用いてもとの問題の解を構成する手間が n に比例するときには，その問題は $\mathrm{O}(n \log n)$ の計算時間で解くことができる．)

問題 12. **procedure** $F(n)$:

if $n = 1$ **then return** A **else**

  if n is even **then** $B \leftarrow F(n/2)$ and **return** B^2

else $B \leftarrow F((n-1)/2)$ and **return** B^2A

A は (したがって B も) 大きさが固定された行列だから，B^2 や B^2A は定数時間で計算できる．上の $F(n)$ の計算時間を $f(n)$ とおくと $f(n) = f(n/2) + c$ (c は定数) である．したがって，$f(n) = \mathrm{O}(\log n)$ である．

問題 13. 数学的帰納法を用いる．(i) $n \le n_0$ に対しては

$$f(n) = c < cn < \frac{cn}{1-(s_1+s_2+\cdots s_k)}$$

である．(ii) 今，$m < n$ を満たすすべての m に対して

$$f(m) \le \frac{cm}{1-(s_1+s_2+\cdots+s_k)}$$

が成り立つと仮定する．このとき，$s_1n, s_2n, s_kn < n$ であるから，

$$\begin{aligned} f(n) &= f(s_1n)+f(s_2n)+\cdots+f(s_kn)+cn \\ &\le \frac{cs_1n+cs_2n+\cdots cs_kn}{1-(s_1+s_2+\cdots s_k)}+cn \\ &= \frac{cn}{1-(s_1+s_2+\cdots s_k)} \end{aligned}$$

が得られる．

(したがって，大きさ n の問題を大きさ $s_1n, s_2n, \cdots, s_kn$ の k 個の小問題に再帰的に分解して解き，小問題の解からもとの問題の解を作ることが n に比例する時間でできる場合には，この解法の時間複雑度は $\mathrm{O}(n)$ となる．)

問題 14. L と M を有向直線とし，それらの左側に凸多角形 A がくるように L と M で A をはさむことを考える．L を反時計回りに回転させて M と一致するまでの回転角を $t(L,M)$ で表す．$t(L,M)=\pi$ のとき，平行線ではさんだことになる．距離最小の平行線ではさんだときには，L と M の少なくとも一方は，A の一辺を含む．このことに注意すると次のアルゴリズムを構成できる．

1. (初期設定) P_1 から P_2 へ向かう有向直線に L を一致させ，P_2 から P_3 へ向かう有向直線に M を一致させる．$\mathrm{P}_1, \mathrm{P}_2, \cdots, \mathrm{P}_n$ の x 座標の最大値と最小値の差を D^* とおく．

2. (反復) 今，L は $\overline{\mathrm{P}_i\mathrm{P}_{i+1}}$ を含み，M は $\overline{\mathrm{P}_j\mathrm{P}_{j+1}}$ を含むとする．

[場合 1]　$t(L,M) < \pi$ なら，M が P_{j+2} を含む位置へくるまで，M を P_{j+1} の回りに反時計回りに回転させる．

[場合 2]　$t(L,M) \ge \pi$ なら，L が P_{i+2} を含む位置へくるまで，L を P_{i+1} の回りに反時計回りに回転させる．

場合 1, 2 のいずれの場合も，この回転の途中で $t(L,M)=\pi$ となる状態を通過したら，そのときの L と M の距離を D とおき，D が D^* より小さければ $D^* \leftarrow D$ とおく．

3. (終了判定) 場合 2 が 1 回以上実行されて，L が再び $\overline{\mathrm{P_1P_2}}$ を含む状態になったら，D^* を出力して処理を終了する．さもなければ 2 へ進む．

このアルゴリズムのステップ 2 は定数時間で実行できる．L が $\overline{\mathrm{P_1P_2}}$ を含む状態から A のまわりをひとまわりして再び $\overline{\mathrm{P_1P_2}}$ を含む状態へ達するまでには，ステップ 2 の場合 2 は n 回実行される．その間にステップ 2 の場合 1 は高々$2n$ 回しか実行されない．したがって，上のアルゴリズムの時間複雑度は $\mathrm{O}(n)$ である．

問題 15. **方針 1**　指定された位置に釘を打つ作業は特に必要ない．x 座標が最大の点 P^* を見つける計算時間は $\mathrm{O}(n)$ である．その点 P^* に結びつけた糸が最初に引っかかる点を見つけるためには，P^* を原点として，他のすべての点を極座標で表し，その偏角の最小値を求めればよい．その計算量は $\mathrm{O}(n)$ である．糸に引っかかる点を一つ見つけるために同様の計算が必要である．凸包の頂点の数は $\mathrm{O}(n)$ だから，全体の時間複雑度は $\mathrm{O}(n^2)$ である．ただし，凸包頂点の数が k のときには $\mathrm{O}(kn)$ であり，k が小さければ，効率は悪くない．

方針 2　x 座標が小さいもの S_1 と大きいもの S_2 のほぼ半分ずつに分け，それぞれの凸包を作ってマージすればよい．ただし,「ほぼ半分ずつ」とは，n が偶数ならちょうど半分ずつ，n が奇数なら個数の差が 1 となるという意味である．マージするためには，両方の凸包に上から接する共通接線と下から接する共通接線を求めなければならないが，それは次のように求めることができる．ここでは上から接する共通接線について述べる．S_1 の x 座標最大の点 P と S_2 の x 座標最小の点 Q を通る直線 L から出発して，S_1 において P の反時計回りの隣りの点が L より上にあれば，P をその点に移し，S_2 において Q の時計回りの隣りの点が L より上にあれば，Q をその点に移すことを L が動かなくなるまでくり返す．最後に得られる L が上から接する共通接線である．これは $\mathrm{O}(n)$ の時間で求められる．下から接する共通接線も同様である．したがって，計算量を $f(n)$ とすると

$$f(n) = 2f\left(\frac{n}{2}\right) + cn$$

と書けるから，$f(n) = \mathrm{O}(n \log n)$ である．

方針 3　すべての点の偏角は $\mathrm{O}(n)$ で計算できる．それらを小さい順に並べることは $\mathrm{O}(n \log n)$ でできる．点の番号を入れ替えて，ここで並べた順を $\mathrm{P}_1, \mathrm{P}_2, \cdots, \mathrm{P}_n$ とする．最後に，そのように並べた点列から，ひっこんでいるところを次の方法で除く．あい続く 3 点の列 $(\mathrm{P}, \mathrm{Q}, \mathrm{R})$ に注目する．最初は $(\mathrm{P}, \mathrm{Q}, \mathrm{R}) \leftarrow (\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3)$ とおく．この 3 点列がひとまわりしてもとの位置にもどるまで，次の場合 1 または場合 2 をくり返す．

[場合 1]　$\mathrm{P}, \mathrm{Q}, \mathrm{R}$ の順に進んだとき，Q で左へ折れるなら 3 点列を一つ先へ進める．すなわち，R の次の点を S とすると，$(\mathrm{P}, \mathrm{Q}, \mathrm{R}) \leftarrow (\mathrm{Q}, \mathrm{R}, \mathrm{S})$ とおく．

[場合 2]　$\mathrm{P}, \mathrm{Q}, \mathrm{R}$ の順に進んだとき，Q で右へ折れるなら Q を削除して，P を一つ前へもどす．すなわち，P の一つ前の点を T とすると，$(\mathrm{P}, \mathrm{Q}, \mathrm{R}) \leftarrow (\mathrm{T}, \mathrm{P}, \mathrm{R})$ とおく．

この手続きが終わったとき，残った点列が凸包頂点の列である．場合 2 を 1 回実行すると点が 1 個減るから，場合 2 は高々n 回しか実行されない．場合 1 を n 回実行

するとひとまわりしてもとの位置へもどる．したがって，この手続きも $\mathrm{O}(n)$ 時間で実行できる．したがって，時間複雑度は $\mathrm{O}(n \log n)$ である．