

実践的幾何アルゴリズム

Advanced Algorithms for Computational Geometry

レポート1と中間試験の
コメント(と解答の一部)
担当: 上原隆平

Advanced Algorithms for Computational Geometry 実践的幾何アルゴリズム

Comments (and part of solutions)
on report 1 and mid-term exam

Ryuhei Uehara

全般的なコメント

- レポートはよくできていました.
 - メールの方はすでに返却しました.
 - 紙の方も返します.
- 試験の得点はけっこう分散しました.

0-9	10-19	20-29	30
0	4	4	1

- 試験の点は個別に問い合わせてください.

Problem 1: アルゴリズム P3-A6 では最大区間和の値を求めた．このとき，区間和の値だけではなく，区間そのもの（始点と終点）を求めるにはどうすればよいか．(In algorithm P3-A6, it computes the maximum value of an interval. In this algorithm, how can you compute not only the maximum value of an interval, but also the start and end indices of the interval?)

Problem 2: (1) コインが 50 円，10 円，5 円，1 円とあったとき，貪欲法で支払うと，支払うコインの枚数が最小になることを証明せよ．(2) さらに 27 円コインがあったときにも貪欲法でコインの枚数が最小になるかどうか考察し，最小になるなら証明し，ならないなら反例を示せ．((1) Assume that we have coins of 50yen, 10yen, 5yen, and 1yen, and use greedy algorithm to pay. Then prove that the number of coins is minimized by the algorithm. (2) Suppose that you have, not only them, but also 27yen coins. Then consider if the greedy algorithm still works. If it does, prove it. If not, show any counterexample.)

(1) は実は 1 円から考えると簡単に証明できる．

Problem 3: Kruskal のアルゴリズムで最小全域木を求めるときと, Prim のアルゴリズムで最小全域木を見つけるときとで, 解が異なることがあるか. ないならそれを証明し, あるならその具体例を示せ. ただし それぞれのアルゴリズムの正当性はすでに証明済みとしてよい. (For finding the minimum spanning tree, is it possible that two solutions by Kruskal's algorithm and Prim's algorithm are different? If the answer is "no", prove it. If the answer is "yes", show an example. You can assume that the correctness of these algorithms has been already proved.)

- アルゴリズムがどちらも正しいのであれば,
 - 解が1つしかなければ, 必ず一致するはず. (誰も気づいてくれなかった ;-)
 - グラフが複数の最小全域木を持つことがあるかどうかを検討すべし.

Problem 4: 配列 $a[0], \dots, a[n-1]$ が与えられたとき, 次の条件を満たす添字の列 $(i_0, i_1, \dots, i_{k-1})$ を「長さ k の単調非減少列」と呼ぶ. (For a given array $a[0], \dots, a[n-1]$, the sequence $(i_0, i_1, \dots, i_{k-1})$ of indices is called a *monotone nondecreasing sequence*.)

- $i_0 < i_1 < \dots < i_{k-1}$
- $a[i_0] \leq a[i_1] \leq \dots \leq a[i_{k-1}]$

与えられた配列 $a[]$ の中で, 最長の単調非減少列を見つけるアルゴリズムを設計し, その実行時間を解析せよ. (For any given array $a[]$, design an algorithm that finds the maximum monotone nondecreasing sequence, and analyse its time complexity.)

応用があるため, よく研究されている.

- $O(n^2)$ のアルゴリズムは, わりと簡単に作れる.
- かなりがんばると $O(n \log n)$ のアルゴリズムにも到達できる
- 相当がんばると $O(n \log \log n)$ のアルゴリズムも作れる

Problem 5: i 番目のフィボナッチ数 F_i は、次のように定義される． (The i th Fibonacci number F_i is defined as follows:)

- $F_0 = F_1 = 1$
- $F_i = F_{i-1} + F_{i-2}$ for $i > 1$

一般の F_n を計算する効率のよいアルゴリズムを設計し、その実行時間を示せ． (Design an efficient algorithm for computing F_n and show its running time.)

- 講義でやったDPの流れで、以下は簡単に作れる
 - 線形時間・線形領域のアルゴリズム
 - 線形時間・定数領域のアルゴリズム
- 行列による表現を使うと $O(\log n)$ 時間で計算できる

$$\begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n$$

- 閉じた式で計算すれば $O(1)$ 時間で計算も可能...

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

Problem 6: 2次元平面上の2つの点集合 $R = \{(1, 2), (2, 1), (3, 1)\}$ と $B = \{(2, 2), (3, 3)\}$ が与えられた時, 線形分離可能問題を線形計画問題を解くことで解け. 特に2つの線形計画問題

$$\begin{array}{ll}
 b \geq -a + 2 & b \leq -a + 2 \\
 b \geq -2a + 1 & b \leq -2a + 1 \\
 b \geq -3a + 1 & b \leq -3a + 1 \\
 b \leq -2a + 2 & b \geq -2a + 2 \\
 b \leq -3a + 3 & b \geq -3a + 3
 \end{array}
 \quad \text{と}$$

の実行可能領域を図示することで実行可能解を持つかどうかをきちんと判定すること. (For given two point sets $R = \{(1, 2), (2, 1), (3, 1)\}$ and $B = \{(2, 2), (3, 3)\}$, solve the linear separability problem by solving linear programs. That is, you have to solve the problem by showing feasible solutions of two linear programs

$$\begin{array}{ll}
 b \geq -a + 2 & b \leq -a + 2 \\
 b \geq -2a + 1 & b \leq -2a + 1 \\
 b \geq -3a + 1 & b \leq -3a + 1 \\
 b \leq -2a + 2 & b \geq -2a + 2 \\
 b \leq -3a + 3 & b \geq -3a + 3.
 \end{array}
 \quad \text{and}$$

[問題 1] 入力の大きさ n に対して, あるアルゴリズムの実行時間 $T(n)$ が $T(n) \leq 2T(n/2) + 3$ で表現できたとする. また $T(1) = 1$ とする. この漸化式を解き, $T(n)$ を n について閉じた式で表せ. (For any input of length n , suppose that the running time $T(n)$ of an algorithm can be represented by $T(n) \leq 2T(n/2) + 3$. We let $T(1) = 1$. Then, solve this recurrence equation and show $T(n)$ by an equation for n .)

[問題 2] 入力の大きさ n に対して, あるアルゴリズムの実行時間 $T(n)$ が $T(n) \leq T(rn) + n^2$ で表現できたとする. ただしここで $0 < r < 1$ は定数とする. この漸化式を解き, $T(n)$ を O 表記で示せ. (For any input of length n , suppose that the running time $T(n)$ of an algorithm can be represented by $T(n) \leq T(rn) + n^2$, where $0 < r < 1$ is a constant. Then, solve this recurrence equation and show $T(n)$ by a big- O notation.)

サービス問題のつもりでしたが...

[問題 3] グラフ $G = (V, E)$ と G 上の 2 頂点 s と t が与えられて、 s と t の間の最短経路長 $\ell(s, t)$ を求めたい。グラフのすべての辺のコストが 1 であるとき、 $\ell(s, t)$ は、次の幅優先探索で計算することができる：

1. 集合 L_0 を $\{s\}$ で初期化する。
2. $i = 0, 1, 2, \dots$, について、 L_i に隣接し、まだ訪問していない頂点をすべて L_{i+1} に追加する。このとき頂点 t を L_{i+1} に加えることになったら、 $\ell(s, t)$ として $i + 1$ を出力して停止する。

ところが辺に正の重みがついていたときは、この単純な方法では正しく $\ell(s, t)$ を求めることができない。具体的にこの方法ではうまくいかない重みつきグラフ G と s, t の例を与えよ。ただし辺の重みは正整数とし、ある経路の長さは、通過した辺の重みの和で定義する。(For a given graph $G = (V, E)$ and two vertices s and t in V , we like to compute the length $\ell(s, t)$ of a shortest path between s and t . If each edge has a unit cost, we can compute $\ell(s, t)$ by the following breadth first search:

1. Initialize a set L_0 by $\{s\}$.
2. For each $i = 0, 1, 2, \dots$, add each unvisited neighbor of a vertex in L_i to L_{i+1} . When the vertex t is added into L_{i+1} , output this $i + 1$ as $\ell(s, t)$, and halt.

However, when each edge has positive weight, this algorithm fails to compute $\ell(s, t)$. Give a counterexample; that is, edge weighted graph G with s and t such that this algorithm fails to compute the length of the shortest path. Here, we assume that each weight of an edge is a positive integer, and the length of a path is defined by the sum of weights of edges on the path.)

• BFSは基本中の基本となるアルゴリズムなので、しっかりとマスターして下さい。

[問題 4]

マージソートの中では、すでに並んでいる長さ n の配列 $a[1], \dots, a[n]$ と、すでに並んでいる長さ n の配列 $b[1], \dots, b[n]$ をマージして、長さ $2n$ のマージしている配列 $c[1], \dots, c[2n]$ を線形時間で作ることが必要であった。この処理を行うアルゴリズムを具体的に示せ。(In merge sort, the algorithm merges two sorted arrays $a[1], \dots, a[n]$ and $b[1], \dots, b[n]$ of length n into one sorted array $c[1], \dots, c[2n]$ of length $2n$ in linear time. Describe the details of this algorithm.)

- 一方をマージしたあと、残りをマージすることを忘れないで欲しかったのですが、そこを忘れていた人も、少しいました。

[問題 5]

3点 $p_1 = (x_1, y_1), p_2 = (x_2, y_2), p_3 = (x_3, y_3)$ が与えられたとき, この3角形 $p_1p_2p_3$ の符号付き面積 S は

$$S = \frac{1}{2}((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)) \quad (1)$$

で与えられる. これを証明せよ. (For any three points $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$, and $p_3 = (x_3, y_3)$, the signed area S of the triangle $p_1p_2p_3$ is given by the equation (1). Prove it.)

- 満点は難しいですが, 部分点をサービスしました.
 - あまり美しい証明が見つからない...