平成27年度

東京大学大学院情報理工学系研究科 コンピュータ科学専攻 入学試験問題

専門科目 I

平成26年8月19日

10:00 – 12:30

注意事項

(1) 試験開始の合図があるまで，この問題冊子を開けないこと．

(2) 4題すべてに答えよ．問題ごとに指定された解答用紙を使用すること．

(3) 解答用紙および問題冊子は持ち帰らないこと．

Specialized Subjects I

10:00 – 12:30, August 19, 2014

Entrance Examination (AY 2015)

Department of Computer Science, Graduate School of Information Science and Technology

The University of Tokyo

Notice:

(1) Do not open this problem booklet until the start of the examination is announced.

(2) Answer the following 4 problems. Use the designated answer sheet for each problem.

(3) Do not take the problem booklet or any answer sheet out of the examination room.

下欄に受験番号を記入すること．

Write your examinee's number in the box below.

受験番号	No.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと．Usable for memos; do not detach.

問題 1

Alice と Bob の二人のプレイヤーの間で行われるゲームを考える．このゲームでは，それぞれのプレイヤーは2つの手 s_1, s_2 のうちひとつを選ぶ．ただし，お互いに相手がどの手を選ぶかは事前にはわからないものとする．

いま，Alice と Bob が選んだ手をそれぞれ S_A, S_B と表し，$S_A = s_i$，$S_B = s_j$ の際に Alice, Bob が獲得する得点をそれぞれ $f_A(s_i, s_j)$，$f_B(s_i, s_j)$ と表す．

以下の問いに答えよ．

(1) $f_A(S_A^*, S_B^*) \geq f_A(S_A, S_B^*)$ が任意の S_A に対して成り立ち，かつ，$f_B(S_A^*, S_B^*) \geq f_B(S_A^*, S_B)$ が任意の S_B に対して成り立つ $S^* = (S_A^*, S_B^*)$ を均衡点とよぶ．

いま，$f_A(s_i, s_j)$ を (i, j) 成分に持つ行列を M_A （すなわち $M_A = \begin{pmatrix} f_A(s_1,s_1) & f_A(s_1,s_2) \\ f_A(s_2,s_1) & f_A(s_2,s_2) \end{pmatrix}$），また $f_B(s_i, s_j)$ を (i, j) 成分に持つ行列を M_B とする．次の (a)，(b) それぞれにおいて，均衡点があればそれらをすべて与えよ．なければその理由を示せ．

(a) $M_A = \begin{pmatrix} 5 & -4 \\ 6 & -3 \end{pmatrix}$，$M_B = \begin{pmatrix} 5 & 6 \\ -4 & -3 \end{pmatrix}$

(b) $M_A = \begin{pmatrix} -1 & 4 \\ 0 & 2 \end{pmatrix}$，$M_B = \begin{pmatrix} -1 & 0 \\ 4 & 2 \end{pmatrix}$

(2) $M_A = \begin{pmatrix} 3 & -2 \\ 2 & 1 \end{pmatrix}$，$M_B = \begin{pmatrix} 2 & -1 \\ 0 & 3 \end{pmatrix}$ で定義されるゲームにおいて，Alice と Bob のそれぞれは，手 s_1 を確率 p, q （ただし $0 \leq p < 1/2$，$0 \leq q \leq 1$）で選択するものとする．

(a) Alice と Bob が獲得する得点の期待値をそれぞれ求めよ．

(b) Bob の獲得する得点の期待値を最大にする q の値を求めよ．

次に，得点が問い (1-a) で定義されるゲームを無限回繰り返す．この際，n 回目のゲームで獲得する得点はもともとの得点の d^{n-1} 倍であるとする（ただし $0 < d < 1$）．プレイヤーは次の 2 つの戦略のうちいずれかをとるものとする．

- （戦略 t_1） 常に s_1 を選ぶ．
- （戦略 t_2） 最初は s_1 を選び，相手が s_1 を選ぶ限り s_1 を選ぶ．相手が s_2 を選んだ次のゲームでは s_2 を選ぶ．一度でも自分で s_2 を選んだら，その後は s_2 を選び続ける．

以下の問いに答えよ．

(3) Alice, Bob が共に戦略 t_1 をとった場合の Bob の総得点 $\tilde{f}_B(t_1, t_1)$ を求めよ．

(4) Alice, Bob が共に戦略 t_2 をとったが，k 回目のゲームで Bob が誤って手 s_2 を選んだ場合を考える．この際の Alice, Bob の総得点をそれぞれ $\tilde{f}_A^*(t_2, t_2)$，$\tilde{f}_B^*(t_2, t_2)$ と表す．

(a) $\tilde{f}_A^*(t_2, t_2)$ を $\tilde{f}_B^*(t_2, t_2)$ と k を用いて表せ．

(b) $\tilde{f}_B^*(t_2, t_2) > \tilde{f}_B(t_1, t_1)$ が成り立つために d がみたすべき条件を求めよ．

Problem 1

Let us consider a game played by two players, Alice and Bob. In this game, each player chooses one out of the two moves s_1 and s_2. We assume that neither player can know the other's choice in advance.

Let S_A and S_B be the moves chosen by Alice and Bob, respectively, and let $f_A(s_i, s_j)$ and $f_B(s_i, s_j)$ denote the scores Alice and Bob obtain, respectively, given that $S_A = s_i$ and $S_B = s_j$.

Answer the following questions.

(1) $S^* = (S_A^*, S_B^*)$ is called an equilibrium if: $f_A(S_A^*, S_B^*) \geq f_A(S_A, S_B^*)$ holds for all S_A, and $f_B(S_A^*, S_B^*) \geq f_B(S_A^*, S_B)$ holds for all S_B.

Now let M_A be the matrix whose (i, j) entry is $f_A(s_i, s_j)$, that is $M_A = \begin{pmatrix} f_A(s_1,s_1) & f_A(s_1,s_2) \\ f_A(s_2,s_1) & f_A(s_2,s_2) \end{pmatrix}$, and M_B be the matrix whose (i, j) entry is $f_B(s_i, s_j)$. For each of the settings (a) and (b) below, find all equilibriums if one exists. If there is none, explain why.

(a) $M_A = \begin{pmatrix} 5 & -4 \\ 6 & -3 \end{pmatrix}$, $M_B = \begin{pmatrix} 5 & 6 \\ -4 & -3 \end{pmatrix}$

(b) $M_A = \begin{pmatrix} -1 & 4 \\ 0 & 2 \end{pmatrix}$, $M_B = \begin{pmatrix} -1 & 0 \\ 4 & 2 \end{pmatrix}$

(2) Consider a game such that $M_A = \begin{pmatrix} 3 & -2 \\ 2 & 1 \end{pmatrix}$ and $M_B = \begin{pmatrix} 2 & -1 \\ 0 & 3 \end{pmatrix}$, and suppose that Alice and Bob choose the move s_1 with probabilities p and q, respectively (here $0 \leq p < 1/2$ and $0 \leq q \leq 1$).

(a) Find the expected score of Alice. Find that of Bob.

(b) Find the value of q that maximizes the expected score of Bob.

We now consider repeating the game defined by Question (1-a) infinitely many times. We define the score obtained in the n-th game to be the original score multiplied by d^{n-1} (here $0 < d < 1$). Each of the players takes one of the following two strategies.

- (Strategy t_1) The player always chooses the move s_1.
- (Strategy t_2) The player initially chooses s_1, and chooses s_1 as long as the other player chooses s_1. If the other player chooses s_2, then the player chooses s_2 in the next game. Once the player chooses s_2, he/she keeps choosing s_2 henceforth.

Answer the following questions.

(3) Find $\tilde{f}_B(t_1, t_1)$, the total score of Bob in case both Alice and Bob take the strategy t_1.

(4) Assume that both Alice and Bob took the strategy t_2, but that in the k-th game Bob chose the move s_2 by accident. Let $\tilde{f}_A^*(t_2, t_2)$ and $\tilde{f}_B^*(t_2, t_2)$ denote the total scores of Alice and Bob in this setting, respectively.

(a) Express $\tilde{f}_A^*(t_2, t_2)$ using $\tilde{f}_B^*(t_2, t_2)$ and k.

(b) Find a condition on d so that we have $\tilde{f}_B^*(t_2, t_2) > \tilde{f}_B(t_1, t_1)$.

問題 2

再帰関数 f を次のように定義する．x は整数とする．

$$f(x) \;=\; \texttt{if}\ x > 35\ \texttt{then}\ x-9\ \texttt{else}\ f(f(x+10)).$$

関数呼び出しの評価は値呼び (call-by-value) 戦略に基づくものとする．以下の問いに答えよ．

(1) $f(35)$ は，以下の簡約列によって 27 に評価される．

$$f(35) \longrightarrow f(f(45)) \longrightarrow f(36) \longrightarrow 27.$$

$f(34)$ の簡約列を示せ．

(2) $x \leq 36$ を満たすすべての整数 x について，$f(x)$ は 27 に評価されることを証明せよ．

(3) $f(x)$ の評価中に関数 f が呼ばれる回数（最初の呼び出し $f(x)$ も含む）を $F(x)$ とする．$F(x)$ が以下の等式を満たすことを示せ．

$$F(x) = \begin{cases} 1 & x > 35 \text{ のとき} \\ 1 + F(x+10) + F(f(x+10)) & x \leq 35 \text{ のとき} \end{cases}$$

(4) 問い (3) の $F(x)$ を x の式で表せ．

Problem 2

We define a recursive function f by:

$$f(x) \;=\; \texttt{if}\ x > 35\ \texttt{then}\ x - 9\ \texttt{else}\ f(f(x+10)) \;,$$

where x is an integer. Assume that function calls are evaluated based on the call-by-value strategy. Answer the questions below.

(1) $f(35)$ is evaluated to 27 by the following reduction sequence.

$$f(35) \longrightarrow f(f(45)) \longrightarrow f(36) \longrightarrow 27.$$

Show the reduction sequence for $f(34)$.

(2) Prove that $f(x)$ is evaluated to 27 for every integer x such that $x \leq 36$.

(3) Let $F(x)$ be the number of calls of the function f during the evaluation of $f(x)$; here we count the initial call of $f(x)$ too. Show that $F(x)$ satisfies the following equations.

$$F(x) = \begin{cases} 1 & \text{if } x > 35 \\ 1 + F(x+10) + F(f(x+10)) & \text{if } x \leq 35 \end{cases}$$

(4) Express $F(x)$ of Question (3) in terms of x.

問題 3

それぞれの棒に重さと長さがある n 個の棒の集合 $B = \{b_1, b_2, \ldots, b_n\}$ を考える．棒 b_i それぞれの重さを w_i，長さを ℓ_i とする．このとき，長さの和を L 以下におさえたまま，B から何本かの棒を選んだ時に，ありえる総重量の最大値を $W_L(B)$ とし，これを求める問題を考える．

以下の問いに答えよ．

(1) 重さがそれぞれ $2, 3, 6, 5, 7$，長さが同じ順にそれぞれ $1, 2, 3, 4, 5$ である棒の集合 B に対して，$W_5(B)$ を求めよ．

(2) すべての長さ ℓ_i および L が正の整数である場合を考える．また，$1 \leq k \leq n$ なる k のそれぞれに対して $B_k = \{b_1, b_2, \ldots, b_k\}$ とおく．何らかの方法で計算した $L+1$ 個の値 $W_j(B_{k-1})$ $(0 \leq j \leq L)$ がすでに得られていると仮定する．このとき，$0 \leq r \leq L$ なる r のそれぞれに対し $W_r(B_k)$ を計算する方法を述べよ．

(3) すべての長さ ℓ_i が正の整数であるとしたときに，整数 L に対して $W_L(B)$ を計算するアルゴリズムを述べ，その時間計算量を n, L を用いて表せ．

(4) すべての長さ ℓ_i が，適当な正の整数の組 c_i, d_i を用いて $\ell_i = c_i + d_i\sqrt{2}$ と表されるとしたときに，整数 L に対して $W_L(B)$ を計算するアルゴリズムを述べ，その時間計算量を n, L を用いて表せ．

Problem 3

Consider a set $B = \{b_1, b_2, \ldots, b_n\}$ of n bars, each of which has its weight and length. Let w_i and ℓ_i denote the weight and the length of the bar b_i, respectively. Consider choosing some bars from B in such a way that the sum of their lengths is not more than L. Let $W_L(B)$ denote the maximum sum of weights among all the possible choices; we are interested in computing $W_L(B)$.

Answer the following questions.

(1) Let the weights of the bars in B be $2, 3, 6, 5$ and 7, and let the lengths of them be $1, 2, 3, 4$ and 5, in the same order. Calculate $W_5(B)$.

(2) Assume that all ℓ_i and L are positive integers. For each k such that $1 \leq k \leq n$, let B_k denote $\{b_1, b_2, \ldots, b_k\}$. Assume that we have already computed, by some means, all of the $(L+1)$ values of $W_j(B_{k-1})$ $(0 \leq j \leq L)$. Describe a method to compute $W_r(B_k)$ in this setting, for each r such that $0 \leq r \leq L$.

(3) Assume that all ℓ_i are positive integers. Describe an algorithm that computes $W_L(B)$ for an integer L, and express its time complexity using n and L.

(4) Assume that each ℓ_i is described as $\ell_i = c_i + d_i\sqrt{2}$, using a certain pair of positive integers c_i and d_i. Describe an algorithm that computes $W_L(B)$ for an integer L, and express its time complexity using n and L.

問題 4

P 台のプロセッサがあり，0 から $P-1$ までの番号がついている．プロセッサ i は，値が格納されている長さ N の整数配列 a_i を持っているとする．これらのプロセッサが互いに通信しながら，それぞれの持つ配列 $a_0, a_1, \ldots, a_{P-1}$ の要素ごとの総和を求め，プロセッサ 0 の配列 a_0 に格納することを考える．つまり，計算前のプロセッサ i の配列を a'_i，計算後のそれを a''_i と書くと，

$$a''_0[j] = \sum_{i=0}^{P-1} a'_i[j] , \qquad j = 0, 1, \ldots, N-1$$

となるようにしたい．以下，これを総和計算とよぶ．

ネットワークに対しては次のような仮定をおく：任意のペア (i_s, i_d) に対し，プロセッサ i_s からプロセッサ i_d に長さ L の整数配列を配送するために $\alpha + \beta L$ の時間がかかる．これは仮にプロセッサ i'_s から i'_d への転送が同時に行われていても，$\{i_s, i_d\} \cap \{i'_s, i'_d\} = \emptyset$ である限り同じ時間がかかるものとする．また，送受信プロセッサが重なる複数の送受信は同時にはできない（早く始まった通信が終了するまで待たされる）ものとする．さらに，整数の和の計算をするのは通信に比べて十分早いのでその所要時間は無視できるものとする．

所要時間は，最初に行われる通信の開始時刻から，プロセッサ 0 上での値がすべて求まるまでの時間とする．

以下の問いに答えよ．

(1) 次のようなアルゴリズムで総和計算をした場合の所要時間を求めよ．

$i = 1, 2, \ldots, P-1$ に対し逐次に以下の処理を行う {
　プロセッサ i がプロセッサ 0 に a_i 全体を送信する．
　プロセッサ 0 に a_i が届いたら，a_i と a_0 の要素ごとの和を a_0 に格納する．
}

(2) P は 2 のべきとし，長さ N は十分短いとする．総和計算を所要時間 $O(\log P)$ で実現するアルゴリズムを考えよ．アルゴリズムを簡潔に説明し，所要時間を示せ．問い (1) で示したアルゴリズムと性能を比べよ．

(3) 次に長さ N は十分大きいものとして，次のアルゴリズムを考える．ここで K はすべてのプロセッサに共通の定数であり，各プロセッサは配列を K 個の部分配列に等分する．

- プロセッサ $P-1$ は，K 個の部分配列を順次プロセッサ $P-2$ に送る．
- プロセッサ i （ただし $1 \le i \le P-2$）は，

  プロセッサ $i+1$ から部分配列が届いたら，対応する自分の部分配列と要素ごとの和をとったうえ，その結果をプロセッサ $i-1$ に送る

  ことを K 回繰り返す．
- プロセッサ 0 は，

  プロセッサ 1 から部分配列が届いたら，対応する自分の部分配列と要素ごとの和をとったうえ，その結果を a_0 に格納する

  ことを K 回繰り返す．

このアルゴリズムの所要時間を示し，このアルゴリズムを最適にする K の値を見積もれ．整数値である必要はない．

(4) 問い (3) のアルゴリズムが，問い (2) で答えたアルゴリズムよりも高速な場合があるかどうか検討せよ．

Problem 4

Assume that there are P processors, indexed from 0 to $P-1$. The processor i has an integer array a_i of length N, filled with values. We would like these processors to communicate with each other, compute the element-wise sum of $a_0, a_1, \ldots, a_{P-1}$, and store the result in the array a_0 of the processor 0. That is: let a_i' denote the array of the processor i before the computation, and let a_i'' denote the array after the computation; we would like

$$a_0''[j] = \sum_{i=0}^{P-1} a_i'[j] \ , \qquad j = 0, 1, \ldots, N-1.$$

This computation is called *reduction* in what follows.

We pose the following assumption on the network: for each pair (i_s, i_d), it takes time $\alpha + \beta L$ to send an integer array of length L from the processor i_s to the processor i_d. This is the case even if another data transfer from the processor i_s' to the processor i_d' takes place concurrently, as long as $\{i_s, i_d\} \cap \{i_s', i_d'\} = \emptyset$. Moreover, no processor can participate in multiple data transfers at the same time (other data transfers are deferred until the first data transfer completes). In what follows, we ignore the time for computing summation, as computation is much faster than communication.

The execution time of reduction is the time from the beginning of the first communication till the moment when the desired sum is stored on the processor 0.

Answer the following questions.

(1) Show the execution time of the following algorithm for reduction.

> Do the following sequentially for $i = 1, 2, \ldots, P-1$ {
>
> the processor i sends the whole array a_i to the processor 0.
>
> the processor 0 receives the array a_i, computes the element-wise sum of a_i and a_0, and stores the sum in a_0.
>
> }

(2) Assume that P is a power of 2 and N is small enough. Give an algorithm that computes reduction in $O(\log P)$ time. Describe the algorithm briefly, and show its execution time. Compare its performance with that of the algorithm in Question (1).

(3) Next assume that N is big enough, and consider the following algorithm. Here K is a constant that is common to all the processors, and the array of each processor is divided into K subarrays of equal sizes.

- The processor $P-1$ sends the K subarrays to the processor $P-2$, one by one.
- The processor i (where $1 \le i \le P-2$) repeats the following, K times:
  When a subarray from the processor $i+1$ arrives, computes the element-wise sum of the subarray and the processor's own corresponding subarray, and sends the result to the processor $i-1$.
- The processor 0 repeats the following, K times:
  When a subarray from the processor 1 arrives, computes the element-wise sum of the subarray and the processor's own corresponding subarray, and stores the sum in a_0.

Show the execution time of the algorithm, and estimate the optimal value of K for this algorithm. The value need not be an integer.

(4) Examine if the algorithm in Question (3) can be faster than your algorithm in Question (2).

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.

余白 (blank page)

計算などに使ってもよいが，切り離さないこと． Usable for memos; do not detach.