２０１６年９月・２０１７年４月入学試験

大学院基幹理工学研究科修士課程

情報理工・情報通信専攻

必修科目（プログラミング）表紙

◎問題用紙が 2 ページあることを試験開始直後に確認しなさい.
◎解答用紙が 2 枚綴りが 1 組あることを試験開始直後に確認しなさい.

【指示】
- この科目は必修科目「プログラミング」である.
- 問題番号 1 と問題番号 2 の 2 題両方について解答すること.
- 問題ごとに指定された解答用紙を 1 枚用いて解答すること.
- 解答にあたっては次の点に注意すること.
  1. プログラムを書く場合には C 言語(C++を含む)の関数(function)または Java 言語の静的メソッド（static method）の定義（definition）の形で書くこと.
  2. 必要に応じて補助の関数やメソッドを定義して使っても良い.
  3. 解答のプログラムの重要部分には日本語または英語で簡潔なコメントを付けること.
  4. 問題文に定められていないことがらを仮定する必要があった場合は，その理由も添えて仮定したことがらを解答中に明記すること.
- 解答のプログラムは，正しさ，効率の良さ，明解さの 3 点に基づいて評価される.

【Directions】
- The subject of this examination is Computer Programming.
- The examination consists of two questions (Questions 1 and 2). Answer both questions.
- Write your answers in the specified answer sheet for each question.
- Read the following instructions before you start:

  1. Programs must be written as C (or C++) functions or Java static methods.
  2. You are allowed to define and use auxiliary functions or methods.
  3. Give brief comments to important parts of your programs.
  4. If you need to assume anything not explicitly mentioned in the problem statement, describe in your answer what you have assumed and why they were necessary.
- Your programs will be evaluated based on their correctness, efficiency, and clarity.

２０１６年９月・２０１７年４月入学試験問題

大学院基幹理工学研究科修士課程情報理工・情報通信専攻

科　目　名：プログラミング

問題番号 1

整数 $0,\ldots,N-1$ $(N>0)$ をいくつか（１個以上Ｎ個以下）のグループ (groups) に分けた状態を，要素数Ｎの 2 本の int 型配列 a と b を使って表現することを考える．ただし配列 a, b の各要素の値は，すべての i, j ($0 \le i < N$, $0 \le j < N$) について以下の 6 個の条件 (a1)〜(b3) を満たすように決める．

(a1) $a[i] \le i$

(a2) $a[i] = j$ ならば i と j は同じグループに属する

(a3) $a[i] = i$ ならば $0,\ldots,i-1$ は i と異なるグループに属する

(b1) $i \ne j$ ならば $b[i] \ne b[j]$

(b2) $b[i] = j$ のとき i と j は同じグループに属する

(b3) i と j が同じグループに属するならば，$b^n[i] = j$ となる $n\ (\ge 0)$ が存在する．ただし $b^n[i]$ は $b^0[i] = i$, $b^{n+1}[i] = b[b^n[i]]$ $(n \ge 0)$ と定義する．

	a	b
[0]	0	9
[1]	1	2
[2]	1	8
[3]	0	0
[4]	1	1
[5]	5	5
[6]	3	3
[7]	7	11
[8]	2	4
[9]	6	10
[10]	3	6
[11]	7	7

たとえば右図の配列 a, b はいずれも {{0,3,6,9,10},{1,2,4,8},{5},{7,11}} というグループ分け (grouping) を表現している．なお，本問のグループ分けにおいては，複数のグループどうしの並べ方（順序）や各グループ内の要素の並べ方は意味をもたないものとする．

(1) 右図の配列 a は，a[1], a[2], a[4], a[8] の値をそれぞれ 1, 1, 1, 2 としてグループ {1,2,4,8} を表現しているが，条件 (a1)〜(a3) を満たす表現方法はほかにもある．グループ {1,2,4,8} を表現する a[1], a[2], a[4], a[8] の値の組 (tuple) を，右図の (1,1,1,2) を含めてすべて列挙せよ．

(2) 配列 a と整数 i, j ($0 \le i < N$, $0 \le j < N$) を受け取り，i と j が同じグループに属する場合は 1，異なるグループに属する場合は 0 を返す関数またはメソッドを書け．

(3) i と j が異なるグループに属しているものとする．a, b, i, j を受け取り，i の属するグループと j の属するグループを一つに併合 (merge) したい（たとえば上記のグループ分けにおいて $i = 6$, $j = 8$ のとき，グループ {5} と {7,11} は変更せずにグループ {0,3,6,9,10} と {1,2,4,8} を併合して {{0,3,6,9,10,1,2,4,8},{5},{7,11}} にしたい）．配列 a と b をそれぞれどのように更新すればよいか，アルゴリズムの概要を説明せよ．なお b の更新は O(1) の時間計算量 (time complexity) で実現可能である．

(4) (3) の作業を行う関数またはメソッドを書け．

(5) グループ分け情報の表現や操作において，配列 a だけでなく配列 b も保持しておくとどのような場合に利点があるかを考えて説明せよ．

(6) 上記のグループ分けの表現法では，ある整数 i が属するグループの要素数 (size) を調べるのに，グループの要素数に比例する手間がかかる．設問 (2)〜(4) の操作の効率を維持したまま，a と b 以外の配列を使わずに，各グループの要素数を高速に調べることができるようにするには，配列 a または b の表現をどのように改善したらよいかを考え，説明せよ．

２０１６年９月・２０１７年４月入学試験問題

大学院基幹理工学研究科修士課程情報理工・情報通信専攻

科　目　名：プログラミング

問題番号 2

XY 座標平面（XY-plane）上に N 個（ただし N≧2）の点があり，もっとも近い 2 点間のユークリッド距離 (Euclidean distance) を求めたい．double 型の一次元配列 px[0], ・・・, px[N-1]および py[0], ・・・, py[N-1] がグローバル変数（static フィールド）として与えられ，px[i]と py[i]（ただし 0≦i<N）が一つの点の X 座標および Y 座標を表すものとする．全ての点の X 座標の値は異なり，Y 座標の値も異なるものとする．また，解答にあたり必要に応じてグローバル変数 (static フィールド) を追加定義し用いて構わない．

(1) N の値が小さいと仮定して，全ての通りを試して解を求め，結果を double 型で返す関数／メソッドを書け．

(2) N の値が大きい場合に全ての通りを試すことは非現実的である．全ての通りを試さずに，(1)よりも計算量の少ない形で解を求めることが期待できる関数／メソッドの設計方針，すなわち，そのアルゴリズムとデータ（変数・配列など）のあらましを説明せよ．大まかな考え方の一つとしては，平面を二つに分割し，それぞれの領域内で最短距離を求めたうえ，領域を超える 2 点間でそれらを下回る距離を探す，という一連の処理を再帰的に行うことが考えられる．

(3) (2)の方針に基づき結果を double 型で返す関数／メソッドを書け．なお，以下の関数／メソッド sort が与えられているものとして用いて構わない．

> void sort(double a[], double b[], int left, int right)　※Java 言語の場合は先頭に static を付与
>
> double 型の同サイズの配列 a および b を受け取り，a の内容をインデックス left から right まで（つまり a[left], ・・・, a[right] まで）昇順でソートする．a のソート時に，a[i]（ただし left≦i≦right）と同じインデックス i の b[i]の値も対応させて動かす．ただし left≧right の場合，left＜0 の場合，あるいは right＋1 が配列のサイズを超える場合，何も処理しない．
> 例えば　a[0]=100, a[1]=5, a[2]=50,　b[0]=0, b[1]=10, b[2]=30, left=0, right=1 を与えて実行すると，実行後には a[0]=5, a[1]=100, a[2]=50,　b[0]=10, b[1]=0, b[2]=30 を得る．ここで配列 a および b のサイズはそれぞれ 3 である．