



バブルソート 2 (Bubble Sort 2)

バブルソートとは、列をソートするアルゴリズムの 1 つである。長さ N の数列 A_0, A_1, \dots, A_{N-1} を昇順にソートしたいとしよう。バブルソートは、隣り合う 2 つの数で大小関係が崩れているものがあれば、それらの位置を交換する。これを、数列を前から順に走査しながら行う。すなわち、 $A_i > A_{i+1}$ となっていれば、その 2 つの数を交換するというのを、 $i = 0, 1, \dots, N-2$ に対してこの順で行うのが 1 回の走査である。この走査を何回か繰り返すことで、数列を昇順にソートできることが知られている。数列 A のバブルソートによる走査回数とは、数列 A に上記のアルゴリズムを適用したときに、数列がソートされるまでに必要な走査の回数である。

JOI 君は今、長さ N の数列 A を持っている。JOI 君はこれから、数列 A の値を変更するクエリを Q 回処理する。具体的には、 $j+1$ 回目 ($0 \leq j \leq Q-1$) のクエリでは、 A_{X_j} の値を V_j に書き換える。

JOI 君は、各クエリを処理した直後の数列について、バブルソートによる走査回数を知りたい。

例 (Example)

長さ $N = 4$ の数列 $A = \{1, 2, 3, 4\}$ があり、クエリの個数は $Q = 2$ とする。クエリの内容は $X = \{0, 2\}$, $V = \{3, 1\}$ とする。

- 1 回目のクエリでは、 A_0 の値を 3 に変更する。 $A = \{3, 2, 3, 4\}$ となる。
- 2 回目のクエリでは、 A_2 の値を 1 に変更する。 $A = \{3, 2, 1, 4\}$ となる。

$A = \{3, 2, 3, 4\}$ に対してバブルソートを適用する。

- まず、 A はソートされていないので、1 回目の走査が開始される。 $A_0 > A_1$ なので、この 2 数を交換し、 $A = \{2, 3, 3, 4\}$ となる。 $A_1 \leq A_2$ なので、この 2 数は交換されない。 $A_2 \leq A_3$ なので、この 2 数は交換されない。
- A がソートされた状態になったので、バブルソートは終了する。

よって、 $A = \{3, 2, 3, 4\}$ のバブルソートによる走査回数は 1 になる。

$A = \{3, 2, 1, 4\}$ に対してバブルソートを適用する。

- まず、 A はソートされていないので、1 回目の走査が開始される。 $A_0 > A_1$ なので、この 2 数を交換し、 $A = \{2, 3, 1, 4\}$ となる。 $A_1 > A_2$ なので、この 2 数を交換し、 $A = \{2, 1, 3, 4\}$ となる。 $A_2 \leq A_3$ なので、この 2 数は交換されない。
- A はまだソートされていないので、2 回目の走査が開始される。 $A_0 > A_1$ なので、この 2 数を交換し、 $A = \{1, 2, 3, 4\}$ となる。 $A_1 \leq A_2$ なので、この 2 数は交換されない。 $A_2 \leq A_3$ なので、この 2 数は交換されない。
- A がソートされた状態になったので、バブルソートは終了する。

よって、 $A = \{3, 2, 1, 4\}$ のバブルソートによる走査回数は 2 になる。



小課題 (Subtasks)

この課題では小課題は全部で 4 個ある。各小課題の配点および制限は以下の通りである。

| 小課題 | 得点 | N | Q | A, V |
|-----|----|--------------------------|--------------------------|-----------------------------------------|
| 1 | 17 | $1 \leq N \leq 2\,000$ | $1 \leq Q \leq 2\,000$ | $1 \leq A_i, V_j \leq 1\,000\,000\,000$ |
| 2 | 21 | $1 \leq N \leq 8\,000$ | $1 \leq Q \leq 8\,000$ | $1 \leq A_i, V_j \leq 1\,000\,000\,000$ |
| 3 | 22 | $1 \leq N \leq 50\,000$ | $1 \leq Q \leq 50\,000$ | $1 \leq A_i, V_j \leq 100$ |
| 4 | 40 | $1 \leq N \leq 500\,000$ | $1 \leq Q \leq 500\,000$ | $1 \leq A_i, V_j \leq 1\,000\,000\,000$ |

実装の詳細 (Implementation details)

Q 個のクエリに答える関数 `countScans` を実装せよ。

- `countScans(A, X, V)`
 - A : 長さ N の整数の配列であり、数列の初期値を表す。
 - X, V : 長さ Q の整数の配列であり、クエリの情報を表す。
 - この関数は長さ Q の整数の配列 S を返さなくてはならない。すべての $0 \leq j \leq Q-1$ に対して、 S_j は $j+1$ 回目のクエリを処理した直後の数列のバブルソートによる走査回数でなくてはならない。

採点プログラムのサンプル (Sample grader)

採点プログラムのサンプルは以下の書式で入力を読み込む。

- 1 行目: $N Q$
- 2 行目: $A_0 A_1 \dots A_{N-1}$
- $3 + j$ 行目 ($0 \leq j \leq Q-1$): $X_j V_j$

採点プログラムのサンプルは以下の書式で `countScans` の戻り値を出力する。

- $1 + j$ 行目 ($0 \leq j \leq Q-1$): S_j