Solution to Intersect

HKOI 2003 Final Senior Question 4

CHAN Siu Man

chan_siuman@acm.org

Hong Kong Olympiads in Informatics

Problem Statement

- Given sets $\mathcal{A}$ and $\mathcal{B}$, find the intersection($\mathcal{A} \cap \mathcal{B}$).

Problem Statement

- Given sets $\mathcal{A}$ and $\mathcal{B}$, find the intersection($\mathcal{A} \cap \mathcal{B}$).
- Sets are specified by intervals.
  Intervals have the form $[a, b]$.
  e.g. $\{1, 3, 4, 5, 7\}$ is specified by $[1, 1], [3, 5], [7, 7]$

Problem Statement

- Given sets $\mathcal{A}$ and $\mathcal{B}$, find the intersection($\mathcal{A} \cap \mathcal{B}$).
- Sets are specified by intervals.
  Intervals have the form $[a, b]$.
  e.g. $\{1, 3, 4, 5, 7\}$ is specified by $[1, 1], [3, 5], [7, 7]$
- Standardized sets:
  - use the minimum number of intervals; and
  - list the intervals in increasing order.

Problem Statement

- Input range:

Problem Statement

- Input range:
  - $1 \leqslant n_1, n_2 \leqslant 1000$
    $n1, n2$ are the numbers of intervals of $\mathcal{A}$ and $\mathcal{B}$.

Problem Statement

- Input range:
  - $1 \leqslant n_1, n_2 \leqslant 1000$
    $n1, n2$ are the numbers of intervals of $\mathcal{A}$ and $\mathcal{B}$.
  - $|a|, |b| \leqslant 10^9$
    all intervals have the form $[a, b]$

Problem Statement

- Input range:
  - $1 \leqslant n_1, n_2 \leqslant 1000$
    $n1, n2$ are the numbers of intervals of $\mathcal{A}$ and $\mathcal{B}$.
  - $|a|, |b| \leqslant 10^9$
    all intervals have the form $[a, b]$
- Scoring:
  For 50% of input, $|a|, |b| \leqslant 10000$.

Solution 1

- Idea: store numbers in boolean arrays; scan and output

Solution 1

- Idea: store numbers in boolean arrays; scan and output
- Complexity: $O(R)$, where R is the range

Solution 1

- Idea: store numbers in boolean arrays; scan and output
- Complexity: $O(R)$, where R is the range
- Expected scoring: 50

Solution 1

- Idea: store numbers in boolean arrays; scan and output
- Complexity: $O(R)$, where R is the range
- Expected scoring: 50
- Data Structure: array

Solution 1

```
INTERSECTION1$(\mathcal{A}, \mathcal{B})$
for $i \leftarrow -10001$ to $10001$
do $Count[i] \leftarrow 0$

for each interval $\mathcal{I}$ of $\mathcal{A}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
    do $Count[j] \leftarrow Count[j] + 1$

for each interval $\mathcal{I}$ of $\mathcal{B}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
    do $Count[j] \leftarrow Count[j] + 1$

for $i \leftarrow -10000$ to $10001$
do if $Count[i] = 2$ and $Count[i-1] \neq 2$
      then STARTNEWINTERVAL$(i)$

    if $Count[i] \neq 2$ and $Count[i-1] = 2$
      then ENDCURRENTINTERVAL$(i-1)$

```

Solution 1

```
INTERSECTION1($\mathcal{A}, \mathcal{B}$)
for $i \leftarrow -10001$ to $10001$
do $Count[i] \leftarrow 0$
     ▷ Clear counter array
for each interval $\mathcal{I}$ of $\mathcal{A}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
     do $Count[j] \leftarrow Count[j] + 1$

for each interval $\mathcal{I}$ of $\mathcal{B}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
     do $Count[j] \leftarrow Count[j] + 1$

for $i \leftarrow -10000$ to $10001$
do if $Count[i] = 2$ and $Count[i-1] \neq 2$
        then STARTNEWINTERVAL($i$)

     if $Count[i] \neq 2$ and $Count[i-1] = 2$
        then ENDCURRENTINTERVAL($i - 1$)

```

Solution 1

```
INTERSECTION1$(\mathcal{A}, \mathcal{B})$
for $i \leftarrow -10001$ to $10001$
do $Count[i] \leftarrow 0$
      ▷ Clear counter array
for each interval $\mathcal{I}$ of $\mathcal{A}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
      do $Count[j] \leftarrow Count[j] + 1$
            ▷ Store all numbers in $\mathcal{A}$ into counter array
for each interval $\mathcal{I}$ of $\mathcal{B}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
      do $Count[j] \leftarrow Count[j] + 1$
            ▷ Store all numbers in $\mathcal{B}$ into counter array
for $i \leftarrow -10000$ to $10001$
do if $Count[i] = 2$ and $Count[i-1] \neq 2$
          then STARTNEWINTERVAL$(i)$

      if $Count[i] \neq 2$ and $Count[i-1] = 2$
          then ENDCURRENTINTERVAL$(i-1)$

```

Solution 1

```
INTERSECTION1($\mathcal{A}, \mathcal{B}$)
for $i \leftarrow -10001$ to $10001$
do $Count[i] \leftarrow 0$
     ▷ Clear counter array
for each interval $\mathcal{I}$ of $\mathcal{A}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
     do $Count[j] \leftarrow Count[j] + 1$
          ▷ Store all numbers in $\mathcal{A}$ into counter array
for each interval $\mathcal{I}$ of $\mathcal{B}$
do for $j \leftarrow \mathcal{I}$.start to $\mathcal{I}$.end
     do $Count[j] \leftarrow Count[j] + 1$
          ▷ Store all numbers in $\mathcal{B}$ into counter array
for $i \leftarrow -10000$ to $10001$
do if $Count[i] = 2$ and $Count[i-1] \neq 2$
        then STARTNEWINTERVAL($i$)
               ▷ A new interval begins at number $i$
     if $Count[i] \neq 2$ and $Count[i-1] = 2$
        then ENDCURRENTINTERVAL($i - 1$)
               ▷ The current interval ends at number $i - 1$
```

Solution 2

- Idea: operate on endpoints of intervals; sort and merge

Solution 2

- Idea: operate on endpoints of intervals; sort and merge
- Complexity: $O(n^2)$ or $O({n_1}^2 + {n_2}^2)$

Solution 2

- Idea: operate on endpoints of intervals; sort and merge
- Complexity: $O(n^2)$ or $O({n_1}^2 + {n_2}^2)$
- Expected scoring: 100

Solution 2

- Idea: operate on endpoints of intervals; sort and merge
- Complexity: $O(n^2)$ or $O({n_1}^2 + {n_2}^2)$
- Expected scoring: 100
- Data Structure: list $\mathcal{L}$

Solution 2

```
INTERSECTION2($\mathcal{A}, \mathcal{B}$)
for each interval $\mathcal{I}$ in $\mathcal{A}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
   Add $\mathcal{I}$.end to $\mathcal{L}$          ▷ Remembers whether it is a start point or end point
for each interval $\mathcal{I}$ in $\mathcal{B}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
   Add $\mathcal{I}$.end to $\mathcal{L}$
Sort $\mathcal{L}$
$C \leftarrow 0$
for $i \leftarrow 0$ to $\mathcal{L}$.length
do if $\mathcal{L}[i]$ is a start point of some interval
      then $C \leftarrow C + 1$
           if $C = 2$
             then STARTNEWINTERVAL($\mathcal{L}[i]$)
      else ▷ $\mathcal{L}$[i] is a end point of some interval
           if $C = 2$
             then ENDCURRENTINTERVAL($\mathcal{L}[i]$)
           $C \leftarrow C - 1$
```

Solution 2

```
INTERSECTION2($\mathcal{A}, \mathcal{B}$)
for each interval $\mathcal{I}$ in $\mathcal{A}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
    Add $\mathcal{I}$.end to $\mathcal{L}$        ▷ Remembers whether it is a start point or end point
for each interval $\mathcal{I}$ in $\mathcal{B}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
    Add $\mathcal{I}$.end to $\mathcal{L}$
Sort $\mathcal{L}$        ▷ Time critical step
$C \leftarrow 0$
for $i \leftarrow 0$ to $\mathcal{L}$.length
do if $\mathcal{L}[i]$ is a start point of some interval
      then $C \leftarrow C + 1$
           if $C = 2$
             then STARTNEWINTERVAL($\mathcal{L}[i]$)
      else  ▷ $\mathcal{L}$[i] is a end point of some interval
           if $C = 2$
             then ENDCURRENTINTERVAL($\mathcal{L}[i]$)
           $C \leftarrow C - 1$
```

Solution 2

```
INTERSECTION2($\mathcal{A}, \mathcal{B}$)
for each interval $\mathcal{I}$ in $\mathcal{A}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
   Add $\mathcal{I}$.end to $\mathcal{L}$        ▷ Remembers whether it is a start point or end point
for each interval $\mathcal{I}$ in $\mathcal{B}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
   Add $\mathcal{I}$.end to $\mathcal{L}$
Sort $\mathcal{L}$        ▷ Time critical step
$C \leftarrow 0$        ▷ Counter for number of overlapping intervals
for $i \leftarrow 0$ to $\mathcal{L}$.length
do if $\mathcal{L}[i]$ is a start point of some interval
      then $C \leftarrow C + 1$
           if $C = 2$
             then STARTNEWINTERVAL($\mathcal{L}[i]$)
      else ▷ $\mathcal{L}$[i] is a end point of some interval
           if $C = 2$
             then ENDCURRENTINTERVAL($\mathcal{L}[i]$)
           $C \leftarrow C - 1$
```

Solution 2

```
INTERSECTION2($\mathcal{A}, \mathcal{B}$)
for each interval $\mathcal{I}$ in $\mathcal{A}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
   Add $\mathcal{I}$.end to $\mathcal{L}$          ▷ Remembers whether it is a start point or end point
for each interval $\mathcal{I}$ in $\mathcal{B}$
do Add $\mathcal{I}$.start to $\mathcal{L}$
   Add $\mathcal{I}$.end to $\mathcal{L}$
Sort $\mathcal{L}$          ▷ Time critical step
$C \leftarrow 0$          ▷ Counter for number of overlapping intervals
for $i \leftarrow 0$ to $\mathcal{L}$.length
do if $\mathcal{L}[i]$ is a start point of some interval
      then $C \leftarrow C + 1$
           if $C = 2$          ▷ start of 2 overlapping intervals
             then STARTNEWINTERVAL($\mathcal{L}[i]$)
      else  ▷ $\mathcal{L}$[i] is a end point of some interval
           if $C = 2$          ▷ end of 2 overlapping intervals
             then ENDCURRENTINTERVAL($\mathcal{L}[i]$)
           $C \leftarrow C - 1$
```

Solution 2

Implementations for sorting:

Solution 2

Implementations for sorting:

- Bubble sort $O(n^2)$

Solution 2

Implementations for sorting:

- Bubble sort $O(n^2)$
- Merge sort $O(n \lg n)$

Solution 2

Implementations for sorting:

- Bubble sort $O(n^2)$
- Merge sort $O(n \lg n)$
- Quick sort $O(n \lg n)$

Solution 2

Implementations for sorting:

- Bubble sort $O(n^2)$
- Merge sort $O(n \lg n)$
- Quick sort $O(n \lg n)$
- Merging of two sorted array $O(n)$

Solution 2

Implementations for sorting:

- Bubble sort $O(n^2)$
- Merge sort $O(n \lg n)$
- Quick sort $O(n \lg n)$
- Merging of two sorted array $O(n)$

Note: All of the above score full marks!

Extensions

What is the solution, if the question:

Extensions

What is the solution, if the question:

- asks for other operations of sets

Extensions

What is the solution, if the question:

- asks for other operations of sets
  - union($\mathcal{A} \cup \mathcal{B}$).

Extensions

What is the solution, if the question:

- asks for other operations of sets
    - union($\mathcal{A} \cup \mathcal{B}$).
    - set difference($\mathcal{A} - \mathcal{B}$).

Extensions

What is the solution, if the question:

- asks for other operations of sets
  - union($\mathcal{A} \cup \mathcal{B}$).
  - set difference($\mathcal{A} - \mathcal{B}$).
  - symmetric difference($\mathcal{A} \oplus \mathcal{B}$).

Extensions

What is the solution, if the question:

- asks for other operations of sets
    - union($\mathcal{A} \cup \mathcal{B}$).
    - set difference($\mathcal{A} - \mathcal{B}$).
    - symmetric difference($\mathcal{A} \oplus \mathcal{B}$).
    - compliment($\bar{\mathcal{A}}$).

Extensions

What is the solution, if the question:

- asks for other operations of sets
    - union($\mathcal{A} \cup \mathcal{B}$).
    - set difference($\mathcal{A} - \mathcal{B}$).
    - symmetric difference($\mathcal{A} \oplus \mathcal{B}$).
    - compliment($\bar{\mathcal{A}}$).
- asks for intersection of multiple sets: $\mathcal{A} \cap \mathcal{B} \cap \mathcal{C} \cap \ldots$

Extensions

What is the solution, if the question:

- asks for other operations of sets
  - union($\mathcal{A} \cup \mathcal{B}$).
  - set difference($\mathcal{A} - \mathcal{B}$).
  - symmetric difference($\mathcal{A} \oplus \mathcal{B}$).
  - compliment($\bar{\mathcal{A}}$).
- asks for intersection of multiple sets: $\mathcal{A} \cap \mathcal{B} \cap \mathcal{C} \cap \ldots$
- allows non-stardardized input.