HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

Algorithms and Data Structures

Laboratory work #4

Michael Kosyakov
Associate Professor

Denis Tarakanov
Assistant

Aglaya Iliina
Associate Professor

hduitmo.ads@yandex.ru

Classes plan

1. Previous homework problem #1444 “Elephpotamus”
2. Problem #1322 “Spy”
3. Task for homework
4. Explanation of test 2

Problem #1444 "Elephpotamus"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Link to the problem's description
  https://acm.timus.ru/problem.aspx?space=1&num=1444&locale=en

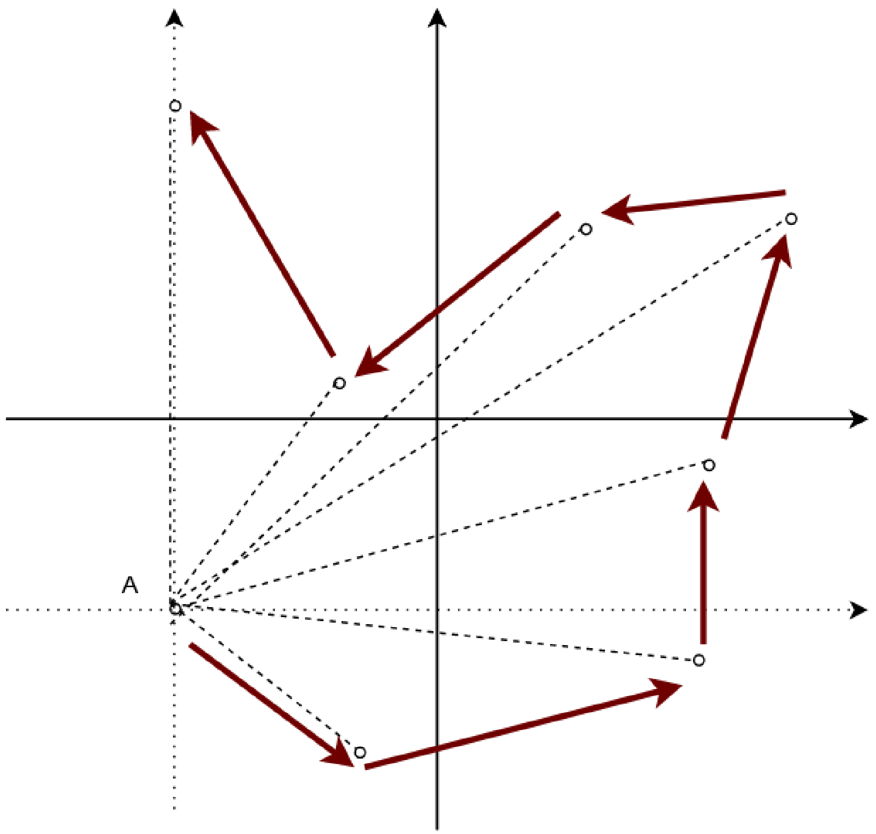
- Elephpotamuses' favorite dainty is elephant pumpkins. Having fed the animal with a pumpkin, Harry can direct it to any of the remaining pumpkins. In order to pass the exam, Harry must lead the elephpotamus so that it eats as many pumpkins as possible before it comes across its footprints.
- It is guaranteed that there are no two pumpkins at the same location and there is no straight line passing through all the pumpkins.
- The first number in this sequence must always be 1.

Problem #1444 “Elephpotamus”

- Another geometric problem
- Median on the Plane
  - Pumpkins are points
  - Start point is center of coordinate plane
  - Need to go through all points in order of angle

Problem #1444 "Elephpotamus"

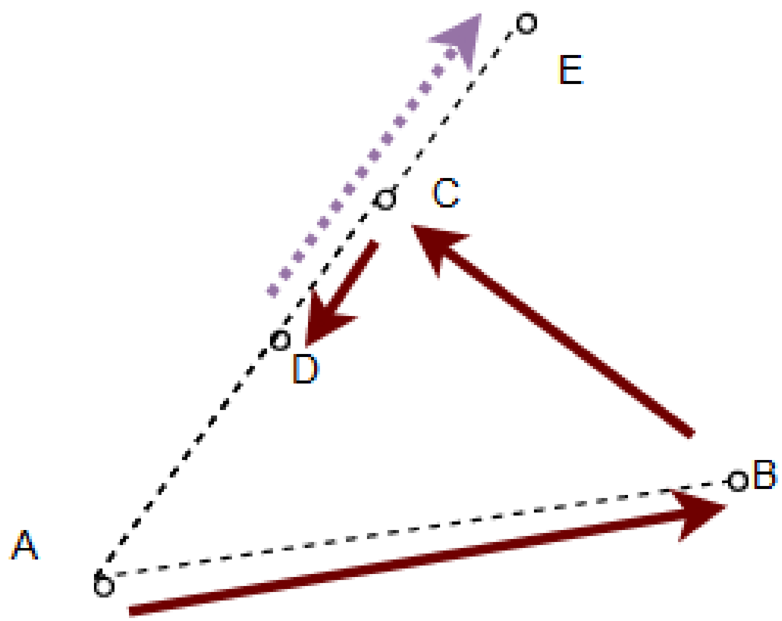
- What the difference?



Problem #1444 "Elephpotamus"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Issue 1. More than 2 points can lie on the same line

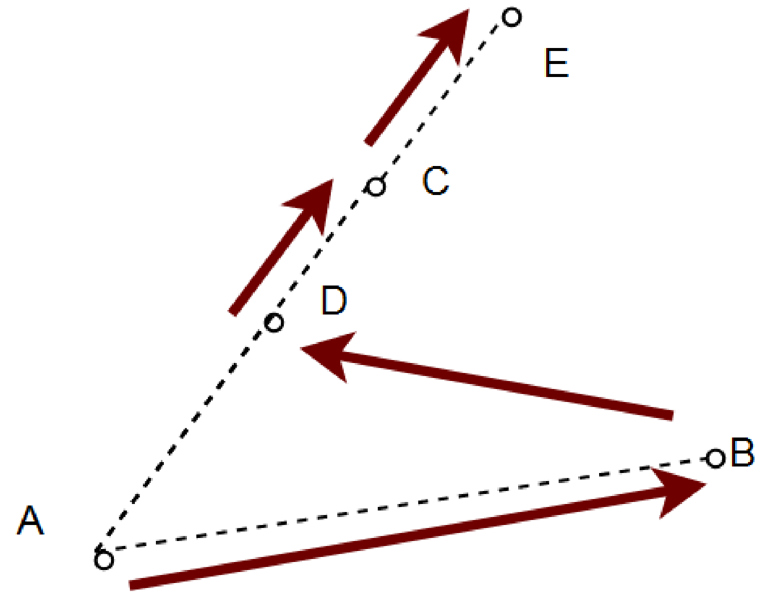


Problem #1444 "Elephpotamus"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Issue 1. More than 2 points can lie on the same line

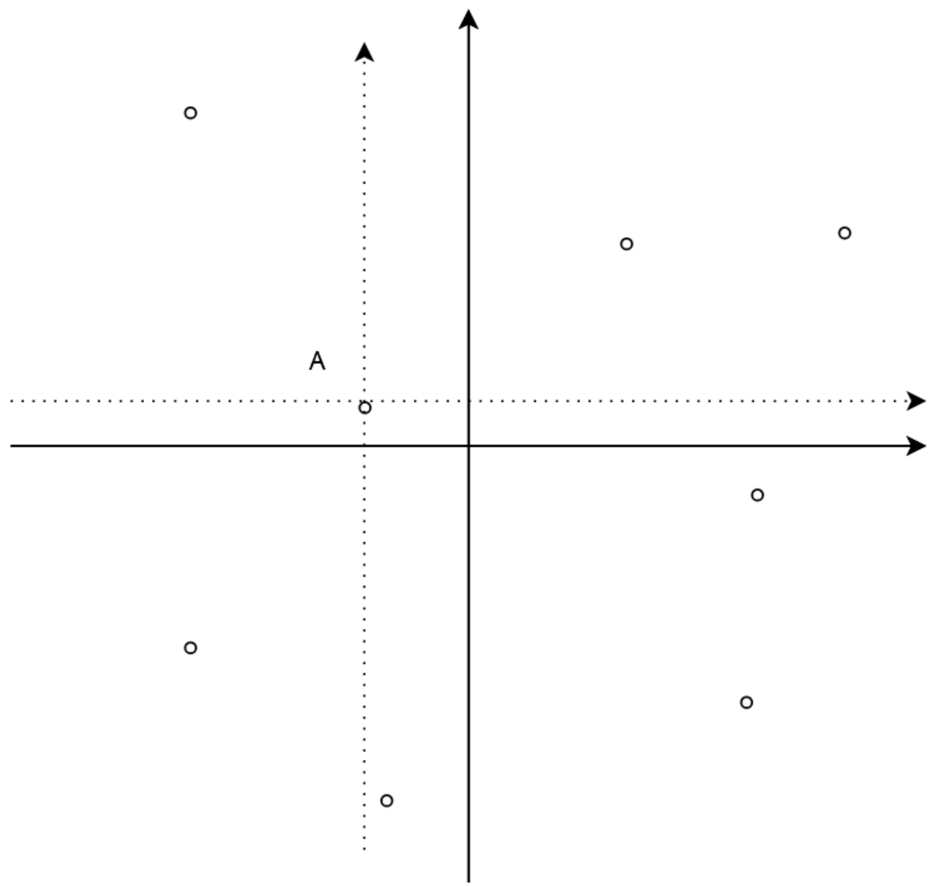
- Second parameter for sorting: distance from start point



Problem #1444 "Elephpotamus"

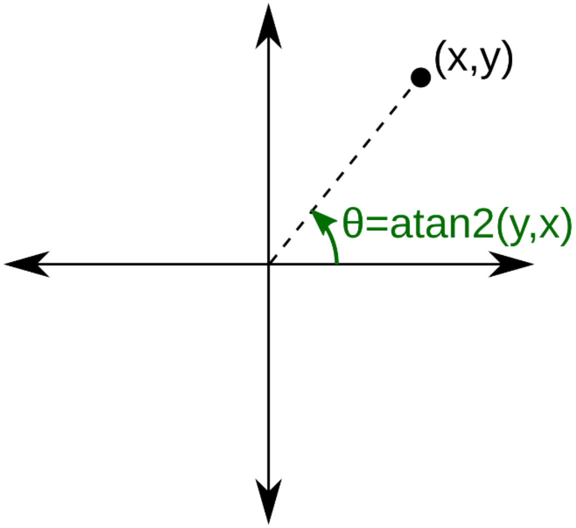
HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

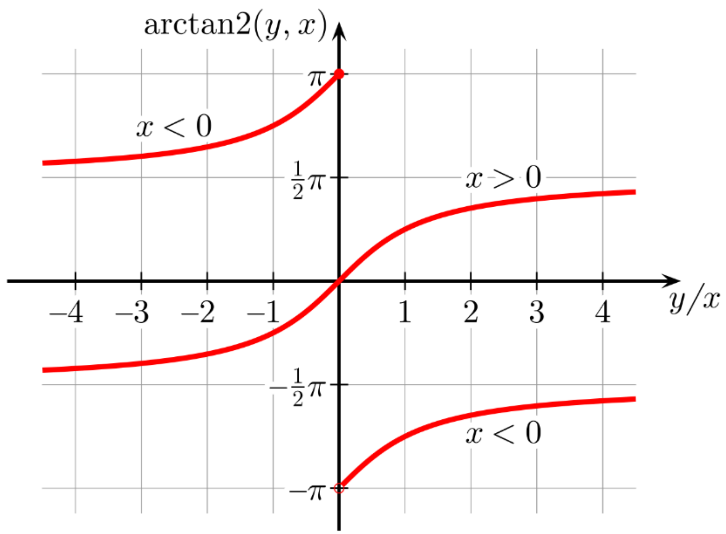
- Issue 2. Can't choose start point



Problem #1444 “Elephpotamus”

- Issue 2. Can’t choose start point
- atan2(y, x) function
  - Return arctangent in range $[-\pi; \pi]$ radians
  - All 360 degrees!

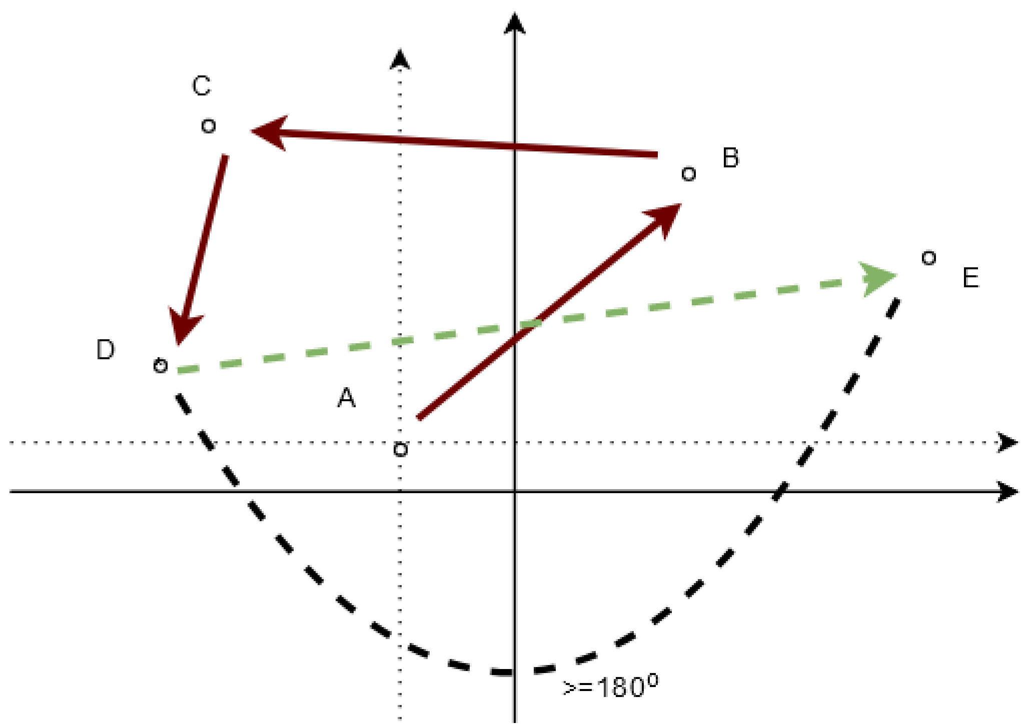




- https://en.cppreference.com/w/cpp/numeric/math/atan2
- https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html#atan2-double-double-

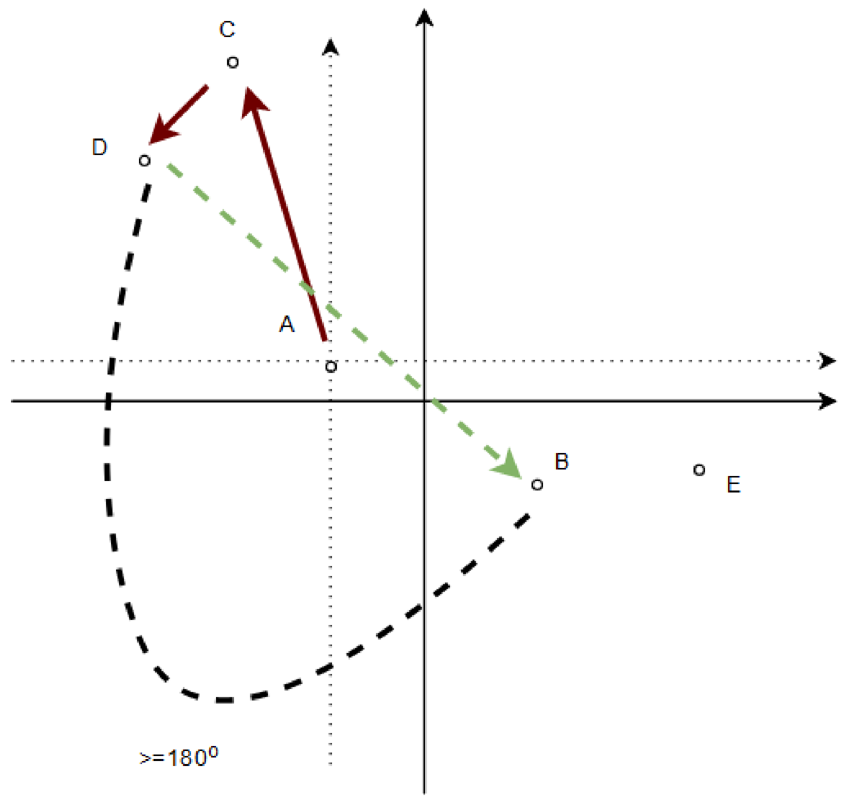
Problem #1444 "Elephpotamus"

- Issue 3. Two consequent points with angle greater or equal to 180 degrees



Problem #1444 "Elephpotamus"

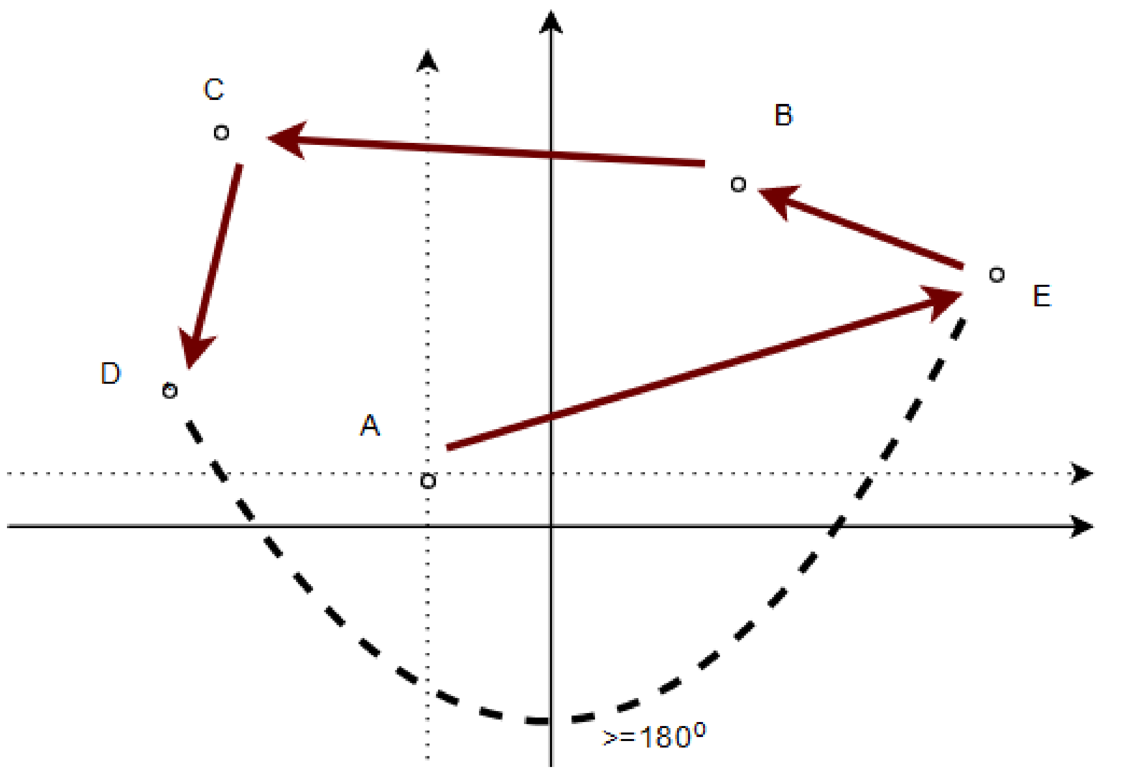
- Issue 3. Two consequent points with angle greater or equal to 180 degrees



Problem #1444 "Elephpotamus"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

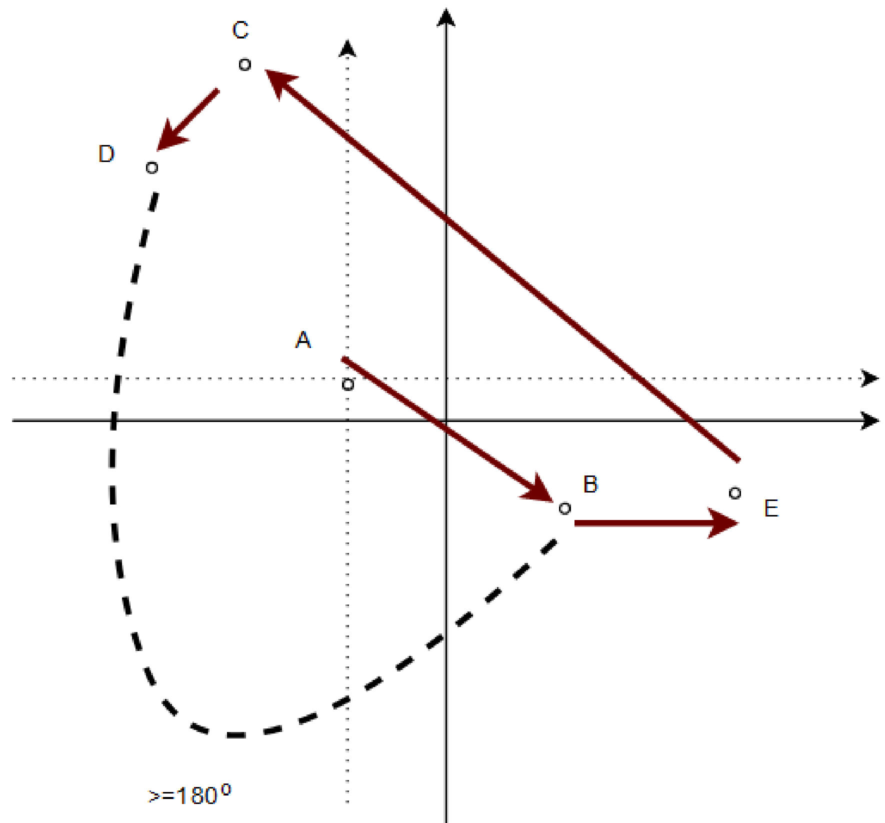
- Solution – detect this case and start from the next point after the gap



Problem #1444 "Elephpotamus"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Solution – detect this case and start from the next point after the gap



Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Link to the problem's description https://acm.timus.ru/problem.aspx?space=1&num=1322&locale=en

- An input of the machine is a text line $S_1 = s_1s_2\ldots s_N$. The machine constructs all cyclic permutations of this line, i.e., $S_2 = s_2s_3\ldots s_Ns_1$, ..., $S_N = s_Ns_1s_2\ldots s_{N-1}$. Then the set S_1, S_2, ..., S_N is sorted lexicographically in the ascending order, and the lines are written out in this order in a column, one under another. Thus, an array $N \times N$ is obtained. One of the rows of this array contains the initial word. The number of this row and the last column of the array are the output of the machine.
- But as the information can certainly be deciphered (otherwise there is no sense in sending it), you have to invent a deciphering algorithm.

Problem #1322 "Spy"

- Burrows-Wheeler transform (BWT)
- Need apply inverse transformation

- Example from Timus:
  - 3
  - rdarcaaaabb

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Naive algorithm: restore full transformation table
  - Step 1. Last column is input string by problem description

										r
										d
										a
										r
										c
										a
										a
										a
										a
										b
										b

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Naive algorithm: restore full transformation table
  - Step 2. Sort this column and add as first column

a										r
a										d
a										a
a										r
a										c
b										a
b										a
c										a
d										a
r										b
r										b

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Naive algorithm: restore full transformation table
  - Step 3. Again, add last column as first and sort pairs

a	…	r
a	…	d
a	…	a
a	…	r
a	…	c
b	…	a
b	…	a
c	…	a
d	…	a
r	…	b
r	…	b

→

r	a
d	a
a	a
r	a
c	a
a	b
a	b
a	c
a	d
b	r
b	r

→

a	a
a	b
a	b
a	c
a	d
b	r
b	r
c	a
d	a
r	a
r	a

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Naive algorithm: restore full transformation table
  - Step 4. Repeat step 2-3 for tuple of 3 characters

a	a									r
a	b									d
a	b									a
a	c									r
a	d									c
b	r									a
b	r									a
c	a									a
d	a									a
r	a									b
r	a									b

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Naive algorithm: restore full transformation table
  - Step 5. Repeat step 2-3 for other characters

a	a	b								r
a	b	r								d
a	b	r								a
a	c	a								r
a	d	a								c
b	r	a								a
b	r	a								a
c	a	d								a
d	a	b								a
r	a	a								b
r	a	c								b

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Naive algorithm: restore full transformation table
  - Step 6. Find resulted line

a	a	b	r	a	c	a	d	a	b	r
a	b	r	a	a	b	r	a	c	a	d
a	b	r	a	c	a	d	a	b	r	a
a	c	a	d	a	b	r	a	a	b	r
a	d	a	b	r	a	a	b	r	a	c
b	r	a	a	b	r	a	c	a	d	a
b	r	a	c	a	d	a	b	r	a	a
c	a	d	a	b	r	a	a	b	r	a
d	a	b	r	a	a	b	r	a	c	a
r	a	a	b	r	a	c	a	d	a	b
r	a	c	a	d	a	b	r	a	a	b

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Time complexity of algorithm: $O(N^3 \log N)$
- Space complexity of algorithm: $O(N^2)$

- Time limit: 0.25 second

- Improvement is required

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Optimized algorithm: vector of inverse transform

a	...	r
a	...	d
a	...	a
a	...	r
a	...	c
b	...	a
b	...	a
c	...	a
d	...	a
r	...	b
r	...	b

→

r	a	...
d	a	...
a	a	...
r	a	...
c	a	...
a	b	...
a	b	...
a	c	...
a	d	...
b	r	...
b	r	...

→

r	a	...
d	a	...
a	a	...
r	a	...
c	a	...
a	b	...
a	b	...
a	c	...
a	d	...
b	r	...
b	r	...

a	a	...
a	b	...
a	b	...
a	c	...
a	d	...
b	r	...
b	r	...
c	a	...
d	a	...
r	a	...
r	a	...

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Optimized algorithm: vector of inverse transform

	i
r	1
d	2
a	3
r	4
c	5
a	6
a	7
a	8
a	9
b	10
b	11

→

j				i
3	a	…	r	1
6	a	…	d	2
7	a	…	a	3
8	a	…	r	4
9	a	…	c	5
10	b	…	a	6
11	b	…	a	7
5	c	…	a	8
2	d	…	a	9
1	r	…	b	10
4	r	…	b	11

→

T
3 -> 1
6 -> 2
7 -> 3
8 -> 4
9 -> 5
10 -> 6
11 -> 7
5 -> 8
2 -> 9
1 -> 10
4 -> 11

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Optimized algorithm: vector of inverse transform
- We have by the task, that target line is 3rd in the table
- The order, from which characters consist 3rd line
- Vector of transform:
  - 3 <- 7 <- 11 <- 4 <- 8 <- 5 <- 9 <- 2 <- 6 <- 10 <- 1 <- 3

T
3 -> 1
6 -> 2
7 -> 3
8 -> 4
9 -> 5
10 -> 6
11 -> 7
5 -> 8
2 -> 9
1 -> 10
4 -> 11

Problem #1322 "Spy"

- Optimized algorithm: vector of inverse transform
  - 3 <- 7(a) <- 11(b) <- 4(r) <- 8(a) <- 5(c) <- 9(a) <- 2(d) <- 6(a) <- 10(b) <- 1(r) <- 3(a)

T 3	7	?	?	?	…
	a	?	?	?	…

T 3	7	11	?	?	…
	a	b	?	?	…

j				i
3	a	…	r	1
6	a	…	d	2
7	a	…	a	3
8	a	…	r	4
9	a	…	c	5
10	b	…	a	6
11	b	…	a	7
5	c	…	a	8
2	d	…	a	9
1	r	…	b	10
4	r	…	b	11

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Optimized algorithm: vector of inverse transform

  - 3 <- 7(a) <- 11(b) <- 4(r) <- 8(a) <- 5(c) <- 9(a) <- 2(d) <- 6(a) <- 10(b) <- 1(r) <- 3(a)

j		i
3	r	1
6	d	2
7	a	3
8	r	4
9	c	5
10	a	6
11	a	7
5	a	8
2	a	9
1	b	10
4	b	11

T 3	7	11	4	8	5	9	2	6	10	1	3
3rd line	a	b	r	a	c	a	d	a	b	r	a

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Further optimizations
  - It is better not to use std::string or String for storing strings
  - Use arrays of type char, which is primitive type
  - Linear time sorting – counting sort

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Counting sort
  - It isn't comparison sort!
  - Limited range of different elements in sorting array (integers, chars, not suitable for continuous numbers)
  - Requires extra memory

Problem #1322 "Spy"

- Counting sort
  - Count how many times each element is presented in the array

Array	5	4	5	3	1	6	7	2
Index	1	2	3	4	5	6	7	8
Count	0	0	0	0	0	0	0	0

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Counting sort

Array	5	4	5	3	1	6	7	2
Index	1	2	3	4	5	6	7	8
Count	0	0	0	0	1	0	0	0

Array	5	4	5	3	1	6	7	2
Index	1	2	3	4	5	6	7	8
Count	0	0	0	1	1	0	0	0

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Counting sort

Array	5	4	5	3	1	6	7	2
Index	1	2	3	4	5	6	7	8
Count	0	0	0	1	2	0	0	0

Result of counting

Array	5	4	5	3	1	6	7	2
Index	1	2	3	4	5	6	7	8
Count	1	1	1	1	2	1	1	0

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Counting sort
  - Restore array from counted values with required sorting

Array	1	2	3	4				
Index	1	2	3	4	5	6	7	8
Count	1	1	1	1	2	1	1	0

Problem #1322 "Spy"

- Counting sort

Array	1	2	3	4	5	5		
Index	1	2	3	4	5	6	7	8
Count	1	1	1	1	2	1	1	0

Sorted array

Array	1	2	3	4	5	5	6	7
Index	1	2	3	4	5	6	7	8
Count	1	1	1	1	2	1	1	0

Problem #1322 "Spy"

HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Counting sort
  - Complexity O(n + k), where k – is number of different indexes in count array
  - Counting sort is not stable!
  - List of elements for each index

Array	R	D	A	R	C	A	A	A	A	B	B
Ordinal index	1	2	3	4	5	6	7	8	9	10	11
Letter	A	B	C	D	R						
Lists	-	-	-	-	-						

Problem #1322 "Spy"

- Stable counting sort

Array	R	D	A	R	C	A	A	A	A	B	B
Ordinal index	1	2	3	4	5	6	7	8	9	10	11
Letter	A	B	C	D	R						
Lists	3			2	1						
					4						

Problem #1322 "Spy"

- Stable counting sort

Array	R	D	A	R	C	A	A	A	A	B	B
Ordinal index	1	2	3	4	5	6	7	8	9	10	11

Letter	A	B	C	D	R
Lists	3	10	5	2	1
	6	11			4
	7				
	8				
	9				

Array	A	A	A	A	A	B	B	C	D	R	R
Ordinal index	3	6	7	8	9	10	11	5	2	1	4

Mandatory task

1. Prepare source code to solve problem #1322 “Spy”
   https://acm.timus.ru/problem.aspx?space=1&num=1322&locale=en
2. Pass tests on Timus system for this problem
   https://acm.timus.ru/submit.aspx?space=1&num=1322
3. Prepare a report with algorithm complexity and explanation
   Use template.docx to prepare report and send it to
   hduitmo.ads@yandex.ru with correct subject

Task for homework

You can solve following problem to get extra 2 points:

1. Problem #1726 “Visits”
   https://acm.timus.ru/problem.aspx?space=1&num=1726&locale=en
   N.B. Report for this problem should contain explanation, why your approach is fast enough

Explanation of test 2

- Task 1
  - Complexity of sorting functions
  - Choose correct statement about Quicksort algorithm

- Correct answers:
  - Quicksort algorithm depends on choice of pivot (partitioning method) and input data
  - Quicksort's complexity: $\Theta(n\log n)$ for best case and degrades till $\Theta(n^2)$ in worst case

Explanation of test 2

- Task 1
  - Complexity of sorting functions
  - Algorithm, which used pair comparisons for sorting, cannot have computational complexity for average-case less than X (but it is possible to have complexity X for this case). What is correct statement for X?

- Correct answers:
  - Comparison sorting algorithms can’t be faster than $O(n \log n)$ for average-case
  - This is a consequence of the limited information available through comparisons alone
  - $O(n \log n)$ can be achieved by such algorithms as merge sort or heap sort

Explanation of test 2

- Task 1
  - Complexity of sorting functions
  - Which algorithm has worst-case time complexity equal to $O(n \log n)$?

- Correct answers:
  - Insertion sort complexity: $O(n^2)$ in worst case
  - Quicksort can degrade till $O(n^2)$ in worst case
  - Heap sort for average and worst case have $O(n \log n)$

Explanation of test 2

- Task 2. Example 1
  - Data structures
  - In data structure elements were inserted in following order:
    { 1, 5, 38, 8, 2, 35, 15 }
    Getting data from this data structure gives following result:
    { 15, 35, 2, 8, 38, 5, 1 }
    What is this data structure?
    - A. Heap
    - B. Stack
    - C. Priority queue (greater elements are prioritized)
    - D. Priority queue (smaller elements are prioritized)

- Correct answer:
  - Elements returned with the principle of LIFO (last in – first out)
  - It is stack

Explanation of test 2

- Task 2. Example 2
  - Data structures
  - In data structure elements were inserted in following order:
    { 1, 5, 38, 8, 2, 35, 15 }
    Getting data from this data structure gives following result:
    { 1, 2, 5, 8, 15, 35, 38 }
    What is this data structure?
    - A. Queue
    - B. Stack
    - C. Linked list (sequential insertion and getting)
    - D. Priority queue (smaller elements are prioritized)

- Correct answer:
  - Elements returned sorted from lower to greater
  - It is priority queue from the least element to greatest

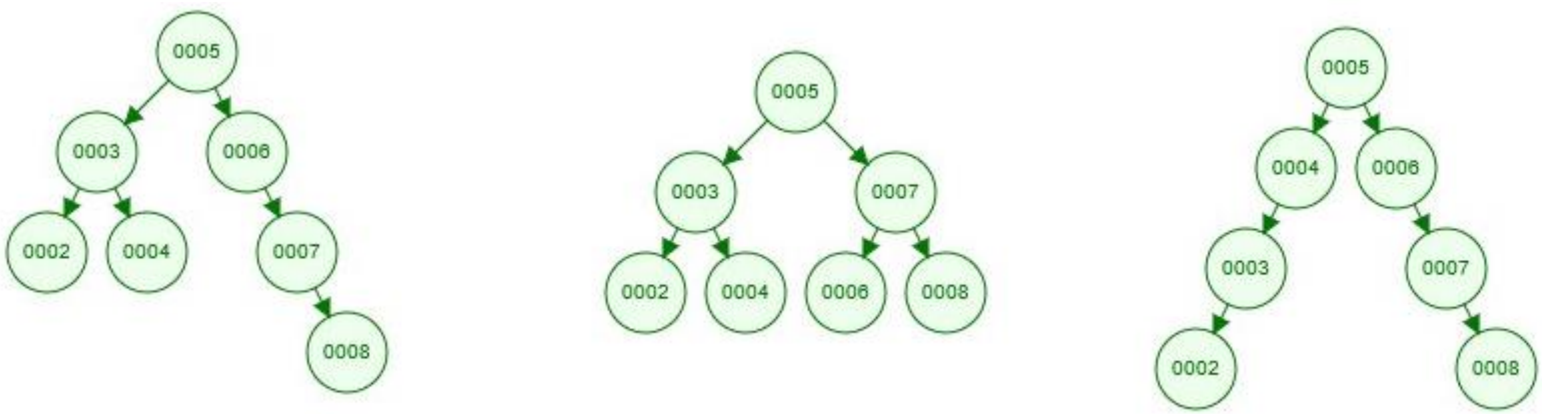
Explanation of test 2

- Task 2. Example 2
  - Data structures
  - In data structure elements were inserted in following order:
    { 1, 5, 38, 8, 2, 35, 15 }
    Examination of memory layout of this data structure gives following result:
    { 38, 15, 35, 5, 8, 1, 2 }
    What is this data structure?
    - A. Stack
    - B. Queue
    - C. Heap
    - D. Array (sequential insertion and getting)

- Correct answer:
  - Question is different!
  - Array, stack and queue is ordered; highest element on top, followed by levels

Explanation of test 2

- Task 3. Example 1
  - Binary search trees
  - Which of these trees are balanced binary search tree?



- Correct answer:
  - All these trees are binary search trees
  - Balanced binary tree is a binary tree in which the height of the left and right subtree of any node differ by not more than 1

Explanation of test 2

- Task 3. Example 2
  - Binary search trees
  - How many comparison operations (in the worst case) do you need to check that element is missed in balanced binary search tree, which has 31 elements (count only comparisons of searched element and keys in balanced binary search tree)?

- Correct answer:
  - Balanced Binary search tree with 31 elements has 5 full levels (1 element on first, 2 on second, 4 on third, etc.)
  - Maximum – 5 comparisons or $\log_2 32$

Explanation of test 2

- Task 3. Example 3
  - Binary search trees
  - How many comparison operations (in the worst case) do you need to check that element is missed in balanced binary search tree, which has 20 elements (count only comparisons of searched element and keys in balanced binary search tree)?

- Correct answer:
  - Balanced Binary search tree with 20 elements has 4 full levels and 1 unfinished (1 element on first, 2 on second, 4 on third, etc.)
  - Maximum – 5 comparisons or $\log_2 32$

Explanation of test 2

- Task 4
- Sorted array vs balanced search tree
  - Select and get operations faster for sorted array: $O(1)$
  - Insert and remove faster for BST: $O(\log N)$
- Linear vs non-linear data types
  - Linear: linked list, queue, deque, stack, array
  - Non-linear: set, map, priority queue, graph
- Bentley-McIlroy and Dijkstra methods for three-way partitioning
  - Dijkstra method: Keeping equal elements near pointer i
  - Bentley-McIlroy: Keeping equal elements near array bounds

Thank you!