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 #6

Michael Kosyakov
Associate Professor

Denis Tarakanov
Assistant

Aglaya Iliina
Associate Professor

hduitmo.ads@yandex.ru

Classes plan

1. Previous homework problem #1067 “Disk Tree”
2. Problem #1628 “White Streaks”
3. Task for homework
4. Explanation of test 3

Problem #1067 "Disk Tree"

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=1067&locale=en

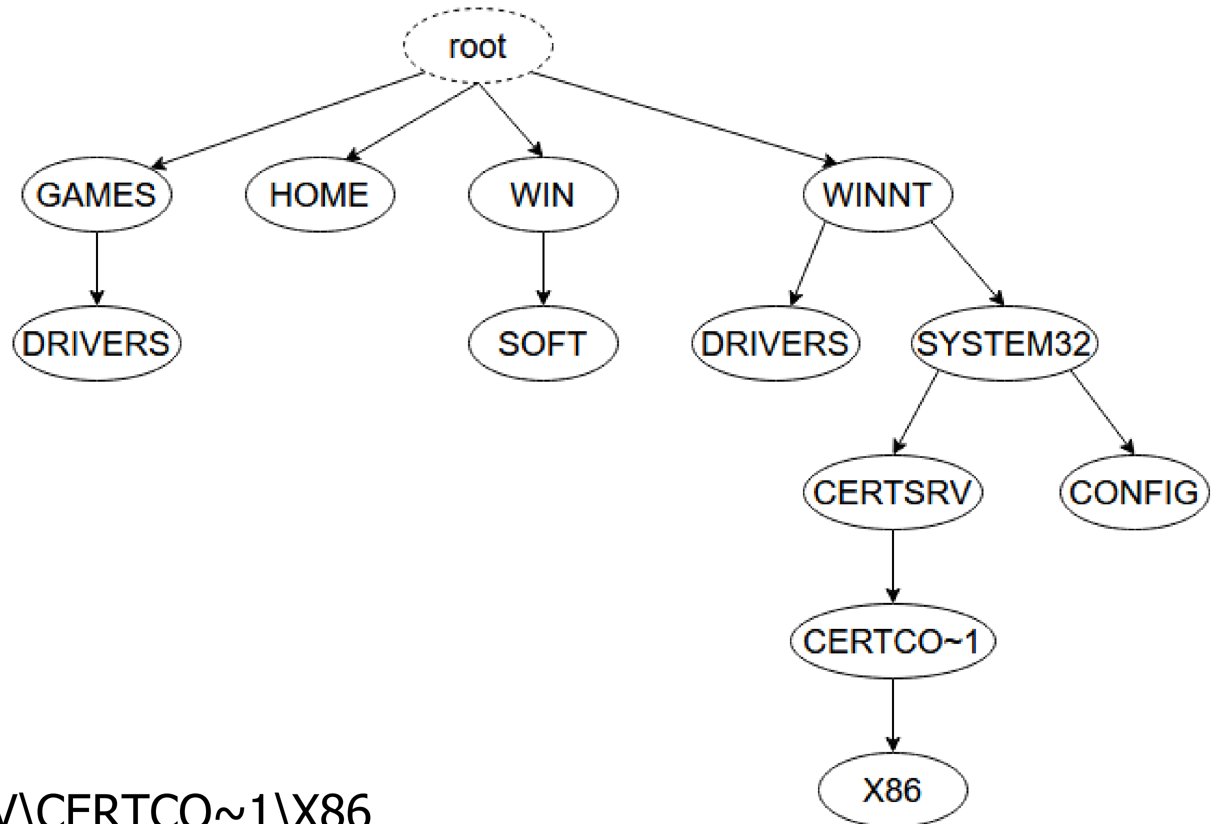
- Bill has several copies of directory listings from his hard drive. Using those listings, he was able to recover full paths (like "WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86") for some directories. He put all of them in a file by writing each path he has found on a separate line.
- Your task is to write a program that will help Bill to restore his state of the art directory structure by providing nicely formatted directory tree.

Problem #1067 "Disk Tree"

- All paths are started from the root and contains only directories
- What is file system?

Problem #1067 "Disk Tree"

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



- Example

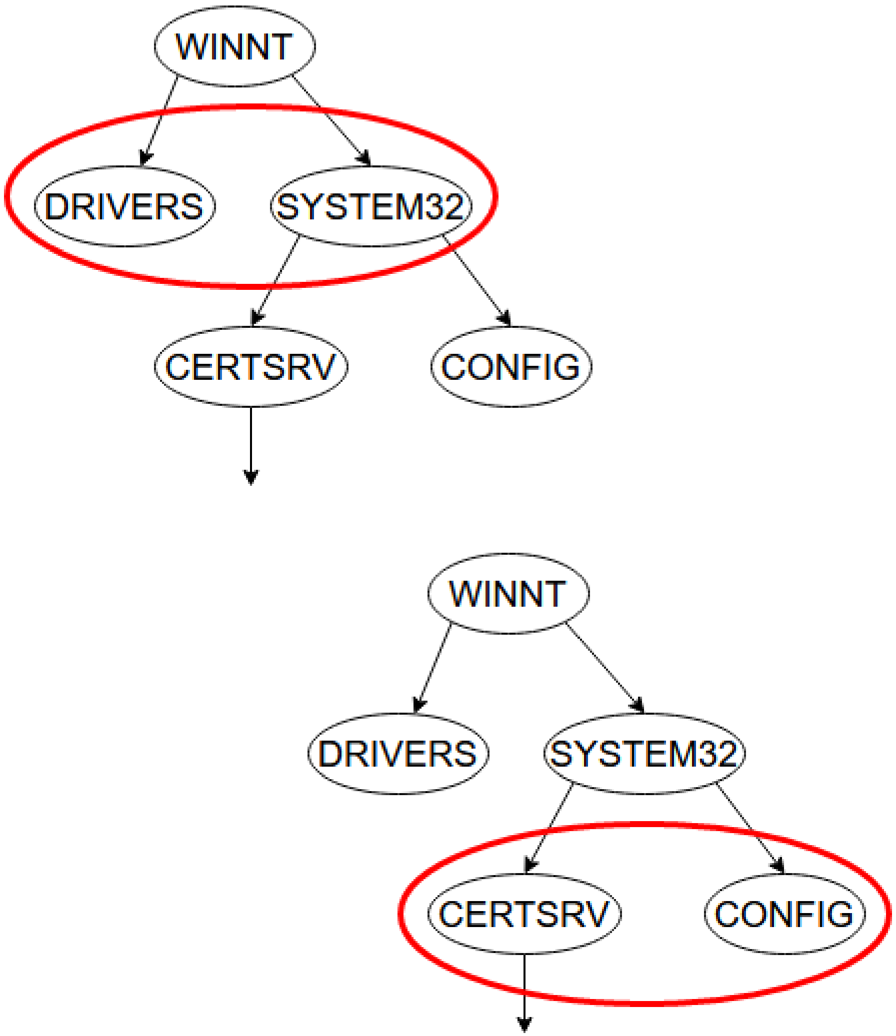
WINNT\SYSTEM32\CONFIG
GAMES
WINNT\DRIVERS
HOME
WIN\SOFT
GAMES\DRIVERS
WINNT\SYSTEM32\CERTSRV\CERTCO~1\X86

WINNT\SYSTEM32\ALGORITHMS

Problem #1067 "Disk Tree"

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

- Each parent node has 0-N children
- Each node can be identified among siblings by its name



Problem #1067
"Disk Tree"

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

- Recursive data structure
- For example
  - Pair of name and vector or list of children
  - Map, where key is the name of node and value is the same recursive data structure

Problem #1067 "Disk Tree"

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

- Reading of paths
  - Split lines by '\'
  - Check if root already has first directory in the path
  - If it has – "open" it by getting element of map with this name and continue for next part in path
  - If it hasn't – add new "directory" to map and continue

- Output can be printed recursively
  - Map has sorted order during iterating (not hash map!)
  - Number of starting spaces depends on current depth of recursion

Problem #1067 "Disk Tree"

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

WINNT\SYSTEM32\CONFIG =>

- WINNT
- WINNT SYSTEM32
- WINNT SYSTEM32 CONFIG

- Sorting approach
- Convert all paths
- Sorting
- Now they're ordered lexicographically
  - If one string starts exactly with another string ("left" part of path) it will be sorting upper
- Remove duplicates
- Iterate and print last word in string
  - Number of spaces depends on number of words in string

Problem #1628 "White Streaks"

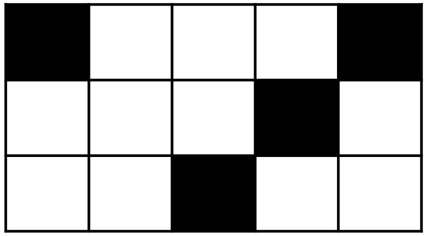
HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Link to the problem's description
  https://acm.timus.ru/problem.aspx?space=1&num=1628&locale=en

- The Martian has a calendar in the form of an $m \times n$ table; he marks in this calendar days when he had bad luck. If he had bad luck in the j^{th} day of the i^{th} week, he paints the cell (i, j) black. Initially, all cells are white.
- Let rectangles of the form $1 \times l$ or $l \times 1$ be called segments of life. Maximal with respect to inclusion white segments are called white streaks. It is required to determine how many white streaks there were in his life.

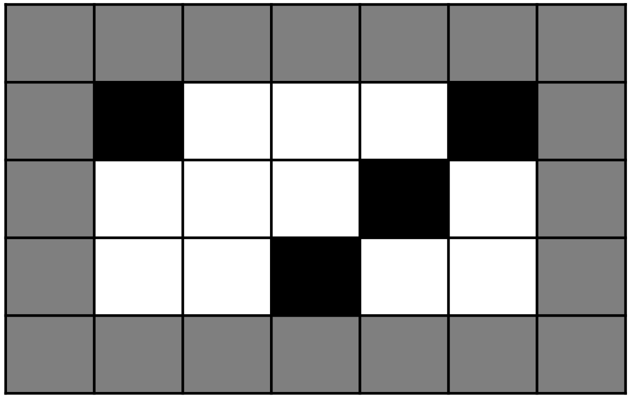
Problem #1628 "White Streaks"

- Need calculate white streaks, which can have size 1×l or l×1



- Calculating separately rows and columns

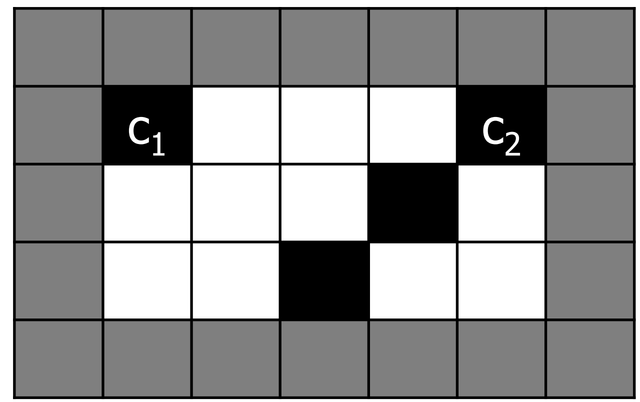
- If representation required, don't forget to add border cells (grey in image)



Problem #1628 "White Streaks"

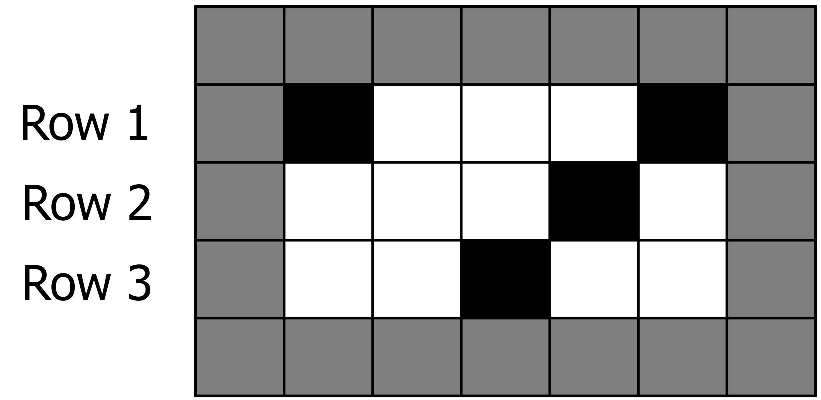
- How to find streaks effectively?

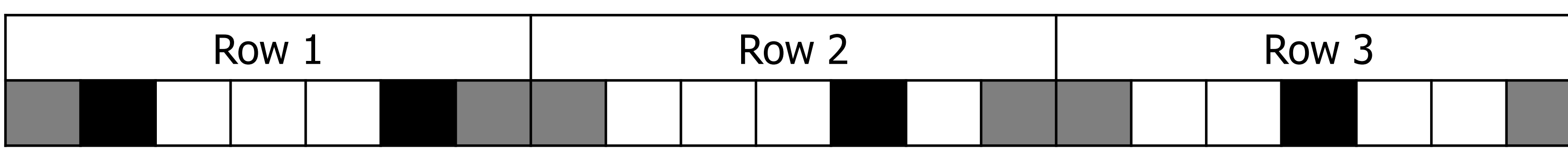
- Cells c_1 and c_2
  - Are they on the same line?
  - Does they have white cells between them?
  - Need to check that there is no black cells between them



Problem #1628 "White Streaks"

- How to effectively find streaks?
  - Sorting black cells by rows
  - Don't forget "grey" cells
  - Calculate difference between consequent black and grey cells positions

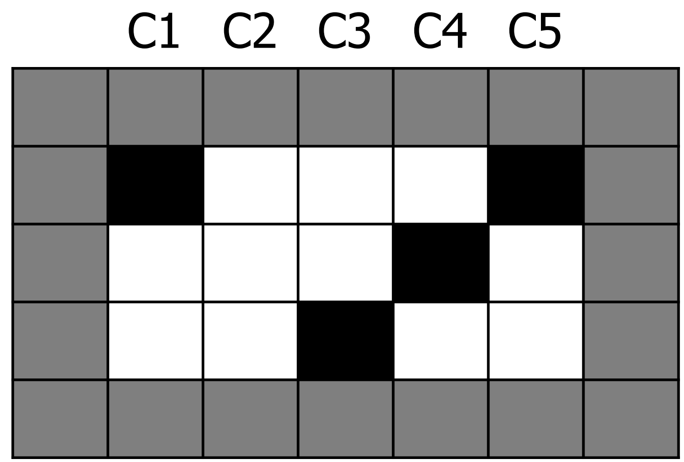


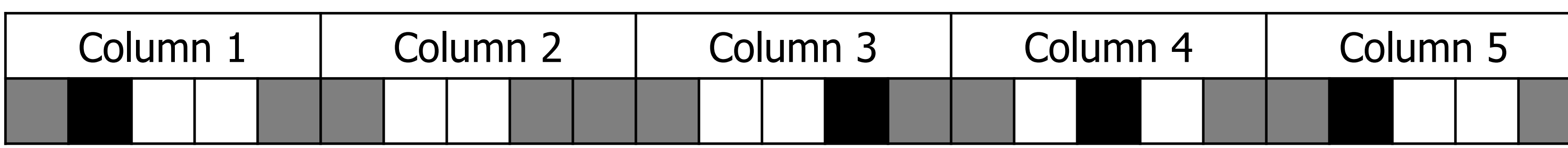


Problem #1628 "White Streaks"

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

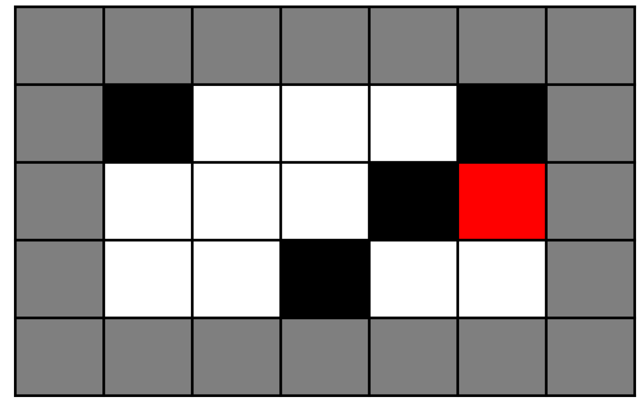
- How to effectively find streaks?
  - Sorting black cells by column
  - Calculate difference between consequent black and grey cells positions

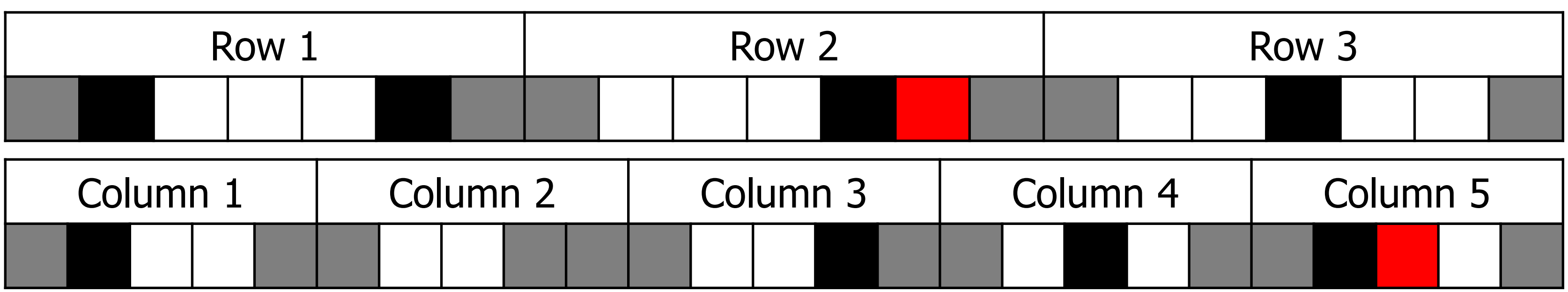




Problem #1628 "White Streaks"

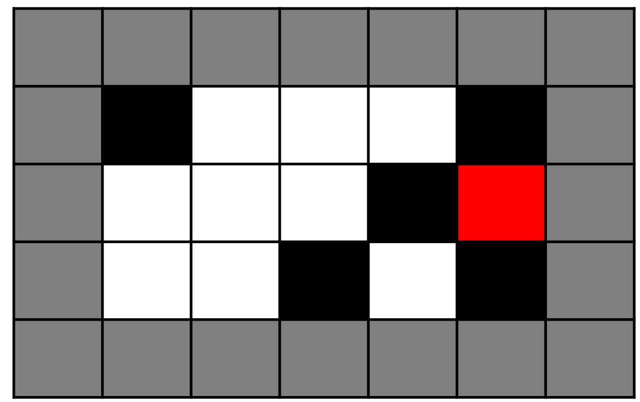
- How to effectively find streaks?
  - 1×1 streaks shouldn't be calculated, if it is part of another streak
  - Check the red cell

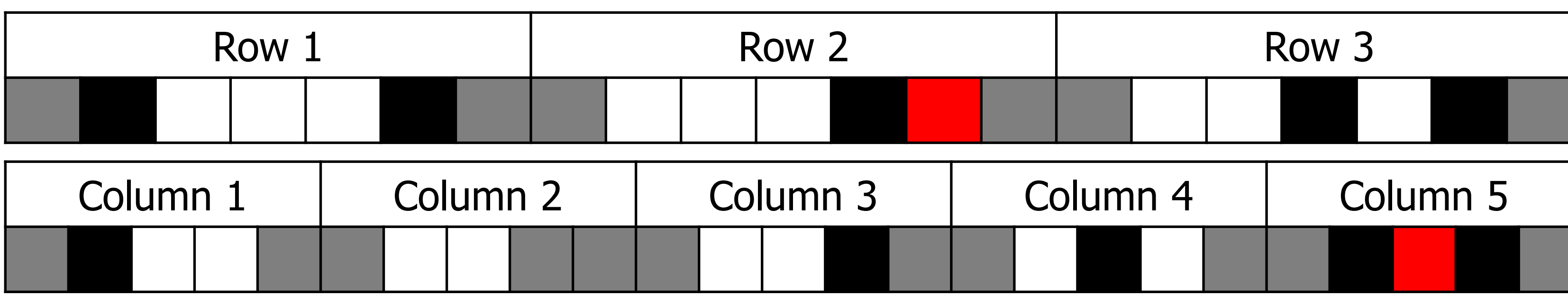




Problem #1628 "White Streaks"

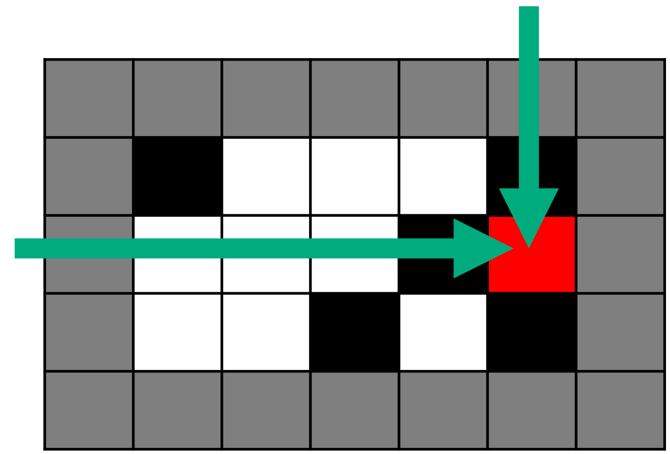
- Possible solution: skip 1×1 streaks for calculating streaks
- Another example
- Red cell isn't calculated!

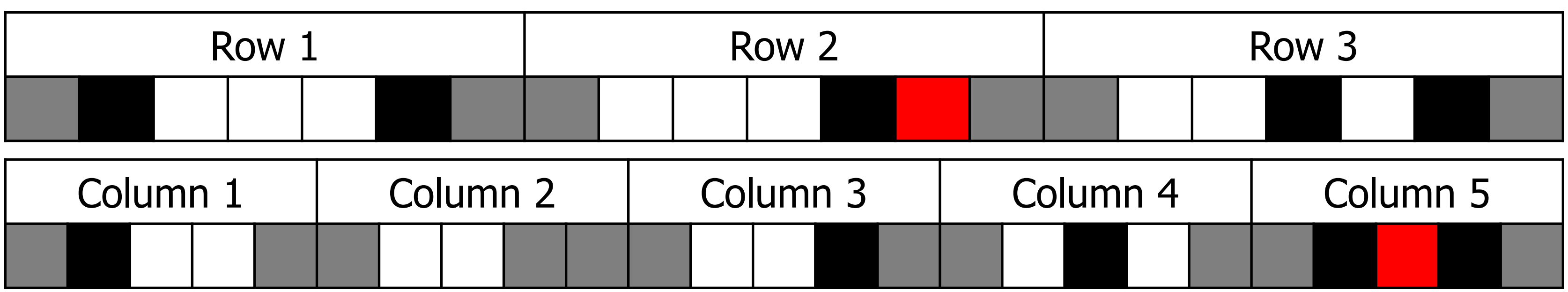




Problem #1628 "White Streaks"

- Better solution:
  - Remember 1×1 streaks during calculating row streaks
  - Check 1×1 streaks during calculating column streaks
  - Only if 1×1 streak was found during both calculations – add it to the sum





Problem #1628 "White Streaks"

- Requires fast data structure for checking 1×1 streaks
  - Set has logarithmic complexity for search

- Resulted formula

$$TotalStreaks = RowStreaks + ColumnStreaks + 1x1CellsInBoth$$

Mandatory task

1. Prepare source code to solve problem #1628 “White Streaks”
   https://acm.timus.ru/problem.aspx?space=1&num=1628&locale=en
2. Pass tests on Timus system for this problem
   https://acm.timus.ru/submit.aspx?space=1&num=1628
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:

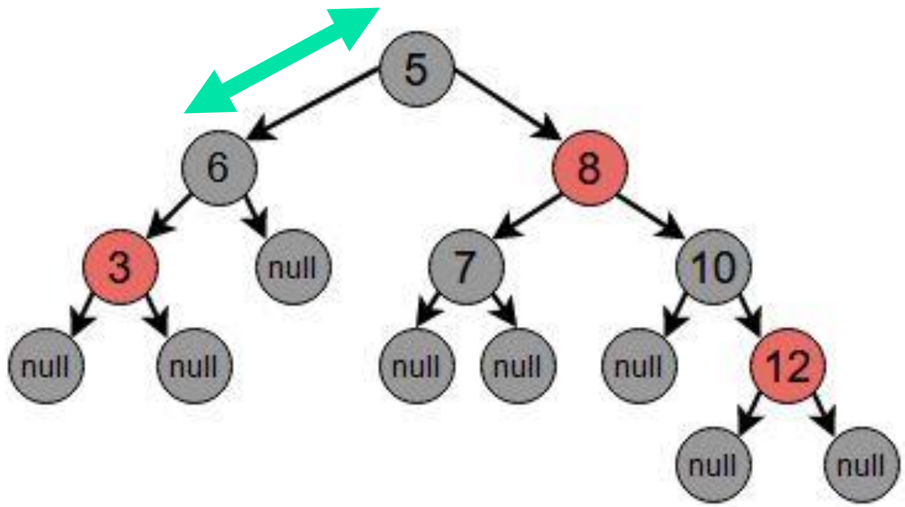
1. Problem #1650 “Billionaires”
   https://acm.timus.ru/problem.aspx?space=1&num=1650&locale=en
   N.B. Report for this problem should contain explanation, which data structures were chosen

Explanation of test 3

- Task 1
- Red-black tree
  - BST that is balanced like a 2-3-4
- Red-black tree rules
  - Every node is either red or black
  - The root and every leaf is black
  - If a node is red, then both its children are black
  - For each node, all paths to descendant leaves contain the same number of black nodes

Explanation of test 3

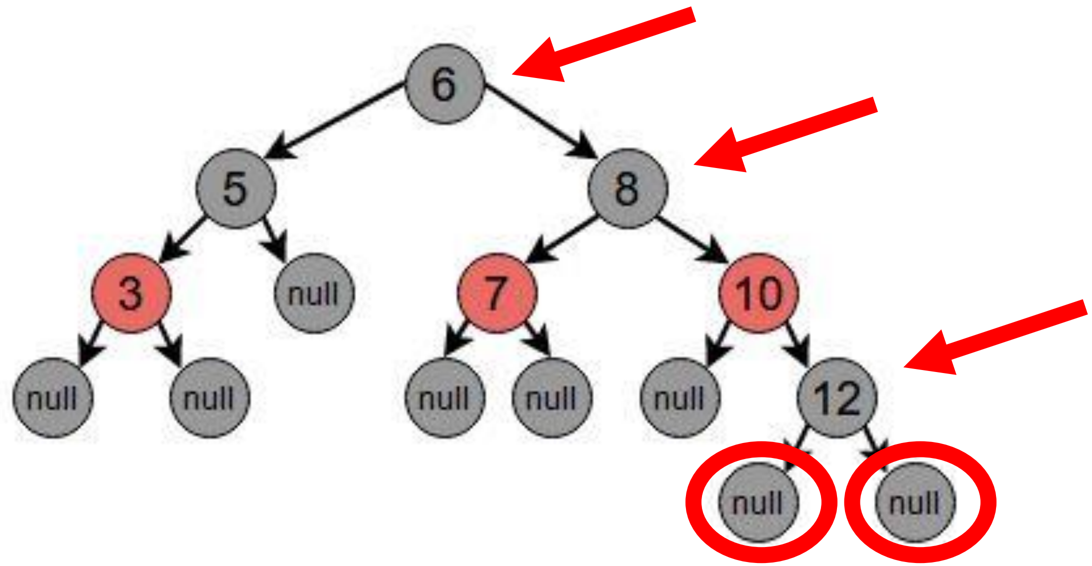
- Task 1. Example 1



- Correct answers:
  - Tree isn’t BST, because left child's value of root is greater, than value of root
  - Swap this nodes

Explanation of test 3

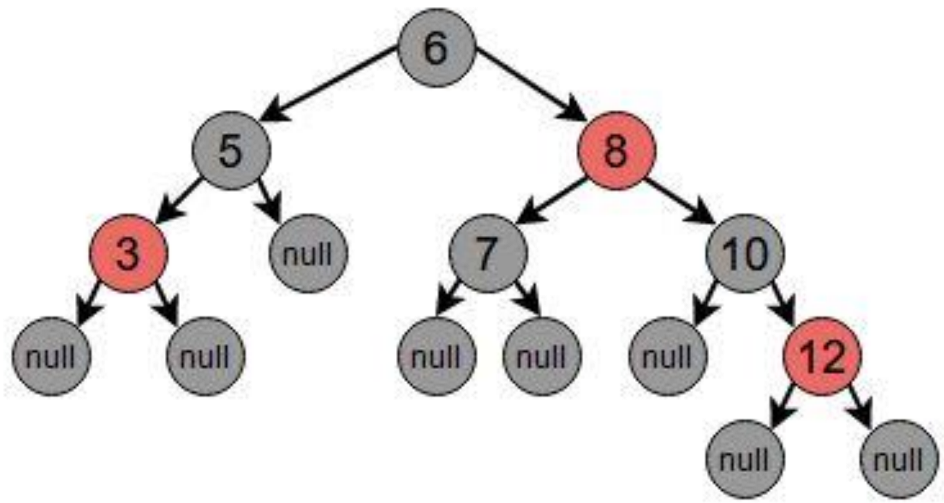
- Task 1. Example 2



- Correct answers:
  - Paths to all descendant leaves contain 2 black nodes, except leaves of node 12 – they have 3 black nodes in path
  - Recolor all non-leaf nodes in right subtree of root

Explanation of test 3

- Task 1. Example 3



- Correct answers:
  - This tree is correct!

Explanation of test 3

- Task 2. Example 1
  - Hash tables and collision resolution
  - In hash table with hash-function "h(k) = k mod 6" and "chaining" method of collision resolution following elements were inserted: { 4, 10, 8, 35, 7 }
    What value(s) physically recorded in cell with index = 4?

index	0	1	2	3	4	5
value	e	e	e	e	e	e

- Insert 4
  - h(4) = 4 mod 6 = 4

index	0	1	2	3	4	5
value	e	e	e	e	4	e

Explanation of test 3

- Task 2. Example 1
  - Hash tables and collision resolution
- Insert 10
  - h(10) = 10 mod 6 = 4
  - Chaining

index	0	1	2	3	4	5
value	e	e	e	e	4, 10	e

- Insert 8
  - h(8) = 8 mod 6 = 2

index	0	1	2	3	4	5
value	e	e	8	e	4, 10	e

Explanation of test 3

- Task 2. Example 1
  - Hash tables and collision resolution
- Insert 35
  - h(35) = 35 mod 6 = 5

index	0	1	2	3	4	5
value	e	e	8	e	4, 10	35

- Insert 7
  - h(7) = 7 mod 6 = 1

index	0	1	2	3	4	5
value	e	7	8	e	4, 10	35

- Correct answer
  - value(4) = 4, 10

Explanation of test 3

- Task 2. Example 2
  - Hash tables and collision resolution
  - In hash table with hash-function "h(k) = k mod 6" and “open addressing with linear probing” method of collision resolution following elements were inserted: { 4, 10, 8, 35, 7 }
    What value(s) physically recorded in cell with index = 5?

index	0	1	2	3	4	5
value	e	e	e	e	e	e

- Insert 4
  - h(4) = 4 mod 6 = 4

index	0	1	2	3	4	5
value	e	e	e	e	4	e

Explanation of test 3

- Task 2. Example 2
  - Hash tables and collision resolution
- Insert 10
  - h(10) = 10 mod 6 = 4
  - Linear probing

index	0	1	2	3	4	5
value	e	e	e	e	4	10

- Insert 8
  - h(8) = 8 mod 6 = 2

index	0	1	2	3	4	5
value	e	e	8	e	4	10

Explanation of test 3

- Task 2. Example 2
  - Hash tables and collision resolution
- Insert 35
  - h(35) = 35 mod 6 = 5

index	0	1	2	3	4	5
value	35	e	8	e	4	10

- Insert 7
  - h(7) = 7 mod 6 = 1

index	0	1	2	3	4	5
value	35	7	8	e	4	10

- Correct answer
  - value(5) = 10

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

Explanation of test 3

- Task 2. Example 3
  - Hash tables and collision resolution
  - In hash table with hash-function h(k) = k mod 4 and "chaining" method of collision resolution following elements were inserted: { 2, 6, 18, 23 }
    What value(s) physically recorded in cell with index = 0?

index	**0**	**1**	**2**	**3**	**4**
value	e	e	e	e	e

- Insert 4
  - h(2) = 2 mod 4 = 2

index	**0**	**1**	**2**	**3**	**4**
value	e	e	2	e	e

Explanation of test 3

- Task 2. Example 3
  - Hash tables and collision resolution
- Insert 6
  - h(6) = 6 mod 4 = 2
  - Chaining

index	0	1	2	3	4
value	e	e	2, 6	e	e

- Insert 18
  - h(18) = 18 mod 4 = 2

index	0	1	2	3	4
value	e	e	2, 6, 18	e	e

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

Explanation of test 3

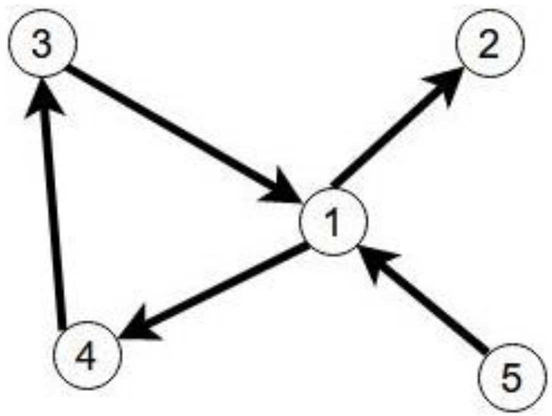
- Task 2. Example 3
  - Hash tables and collision resolution
- Insert 23
  - h(23) = 23 mod 4 = 3

index	0	1	2	3	4
value	e	e	2, 6, 18	23	e

- Correct answer
  - value(0) = empty cell

Explanation of test 3

- Task 3. Example 1
  - Graph representation (adjacency matrix)
  - In this task both outgoing and incoming (with negative values) edges are marked in the adjacency matrix

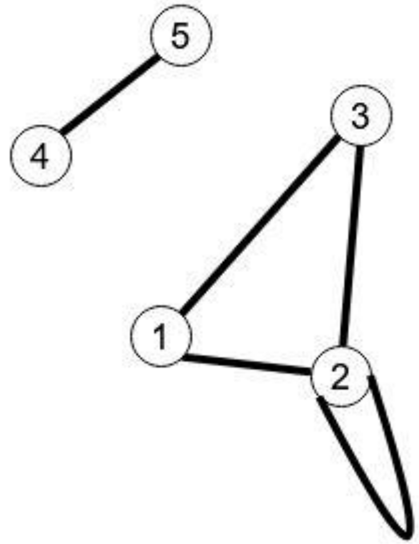


- Directed graph without weights
- Correct answer:

	1	2	3	4	5
1	0	1	-1	1	-1
2	-1	0	0	0	0
3	1	0	0	-1	0
4	-1	0	1	0	0
5	1	0	0	0	0

Explanation of test 3

- Task 3. Example 2
  - Graph representation (adjacency matrix)



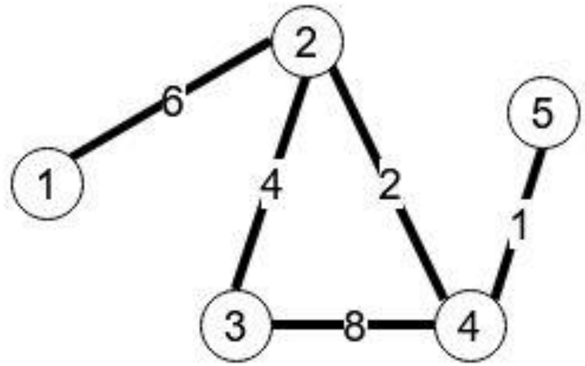
- Undirected graph without weights, loop for vertex 2
- Correct answer:

	1	2	3	4	5
1	0	1	1	0	0
2	1	1	1	0	0
3	1	1	0	0	0
4	0	0	0	0	1
5	0	0	0	1	0

Explanation of test 3

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

- Task 3. Example 3
  - Graph representation (adjacency matrix)



- Undirected graph with weights
- Correct answer:

	1	2	3	4	5
1	0	6	0	0	0
2	6	0	4	2	0
3	0	4	0	8	0
4	0	2	8	0	1
5	0	0	0	1	0

Explanation of test 3

- Task 4
- Hash tables vs Binary search trees
  - Hash table iteration is slower than BST traversal
  - Main operations on balanced BST are guaranteed to run in O(log N), hash table operations are unbounded
- Adjacency matrix vs adjacency list
  - Matrix: memory – $\Theta(n^2)$, check edge between vertices – $O(1)$
  - List: memory – $\Theta(m)$, check edge between vertices – $O(n)$
- 2-3-4 tree vs Binary search tree
  - B-trees – *self-balancing* search trees which allow nodes to have more than one key
  - 2-3-4 trees – a subset of B-trees where each node can have two, three or four children
  - Height of unbalanced BST can be an issue

Thank you!