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 #5

Michael Kosyakov
Associate Professor

Denis Tarakanov
Assistant

Aglaya Iliina
Associate Professor

hduitmo.ads@yandex.ru

Classes plan

1. Previous homework problem #1726 “Visits”
2. Problem #1521 “War Games 2”
3. Problem #1494 “Monobilliards”
4. Task for homework

Problem #1726 “Visits”

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

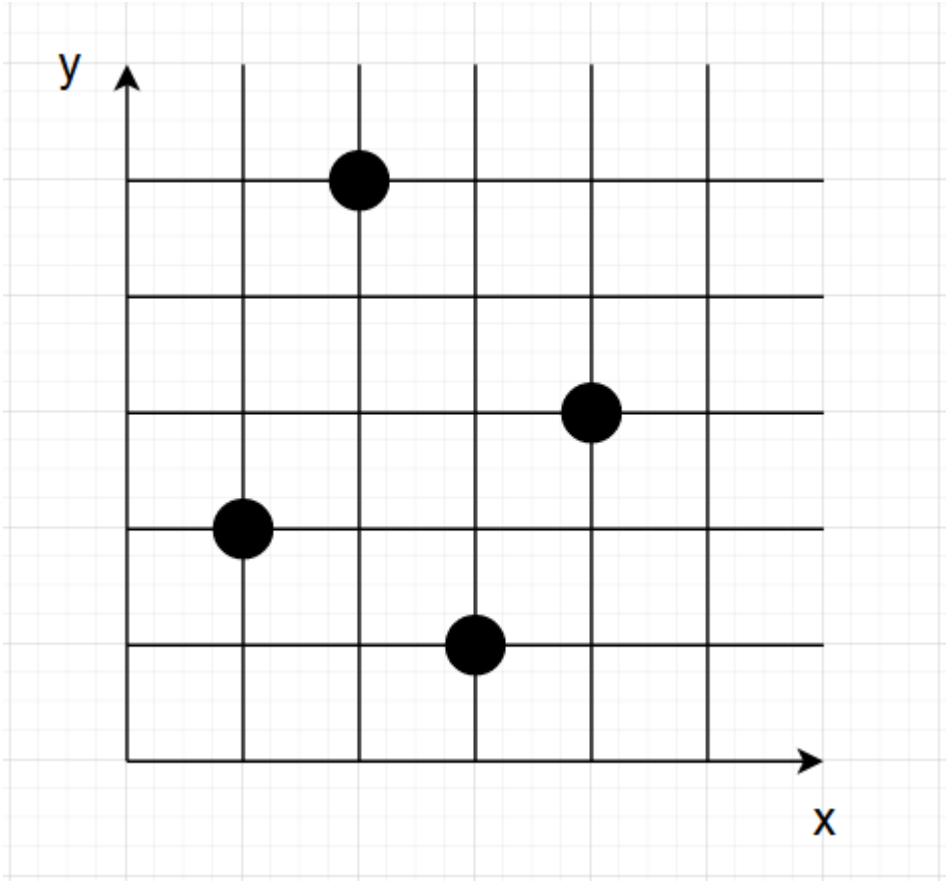
- Link to the problem's description https://acm.timus.ru/problem.aspx?space=1&num=1726&locale=en

- The city is a rectangle with the sides parallel to the coordinate axes. All the streets stretch from east to west or from north to south through the whole city. The house of each member of the program committee is located strictly at the intersection of two orthogonal streets. When walking from one house to another, members of the program committee always choose the shortest way and walk only along the streets. All of them visit each other equally often.
- Output is the average distance, rounded down to an integer, that a member of the program committee walks from his house to the house of his colleague.

Problem #1726 “Visits”

- City is coordinate plane
- Streets are parallel
- Output is average distance between each houses

- Taxicab geometry (Manhattan distance)



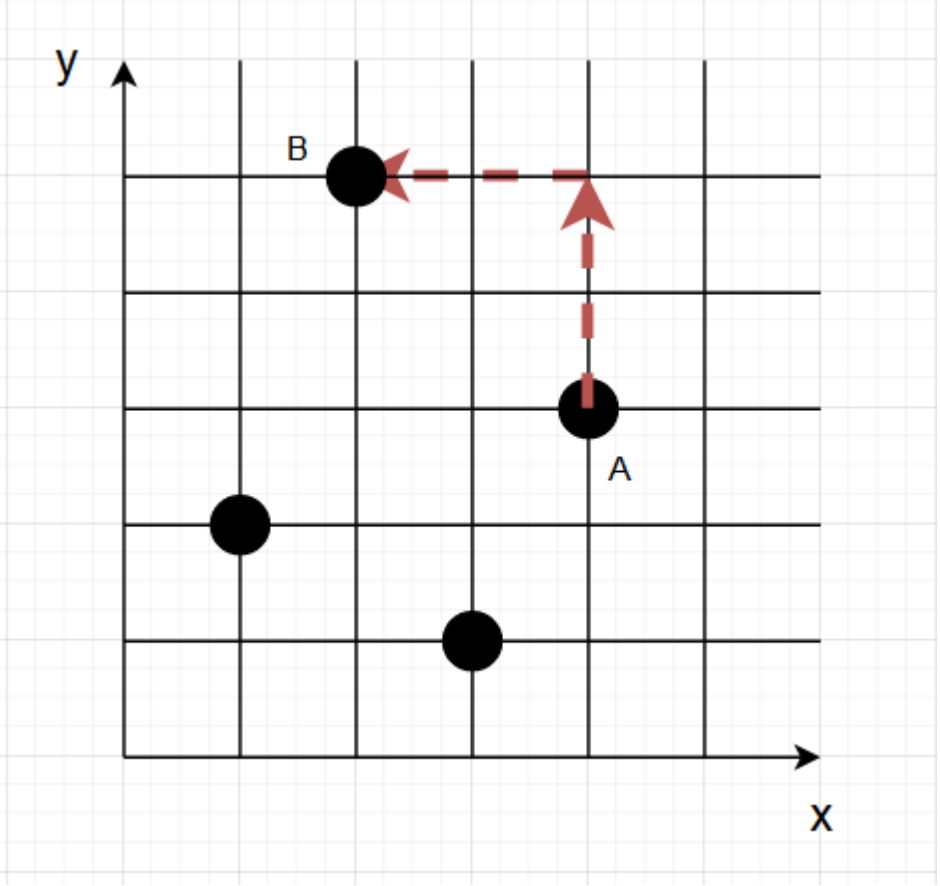
Problem #1726 "Visits"

- Distance between points A and B

$$S_{AB} = |x_A - x_B| + |y_A - y_B|$$

- How many pairs?

$$P = \frac{n(n-1)}{2} = O(n^2)$$



Problem #1726 "Visits"

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

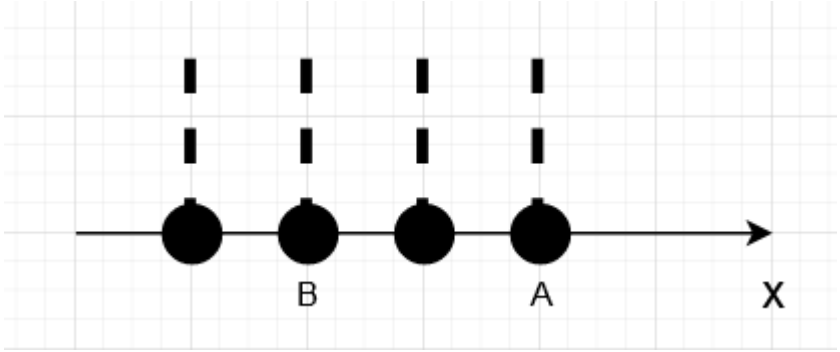
- Total formula:

$$Avg = \frac{\sum_{i=1}^{n-1} \sum_{j=j+1}^{n} (|x_i - x_j| + |y_i - y_j|)}{P}$$

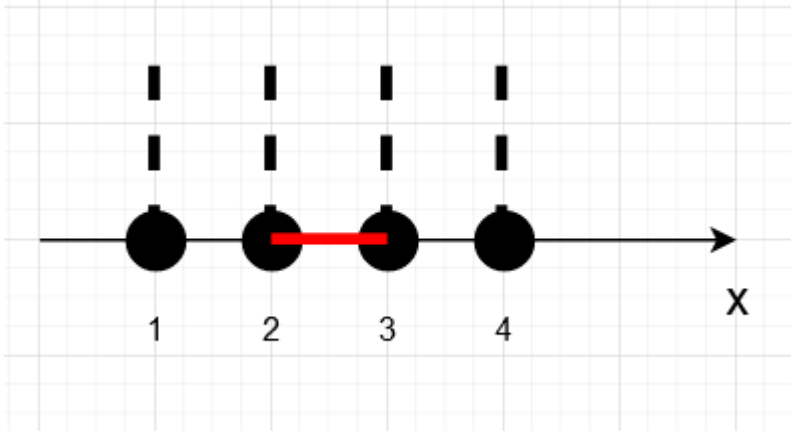
- Paths parallel Ox and parallel Oy can be calculated separately

Problem #1726 “Visits”

- Projection on axis Ox

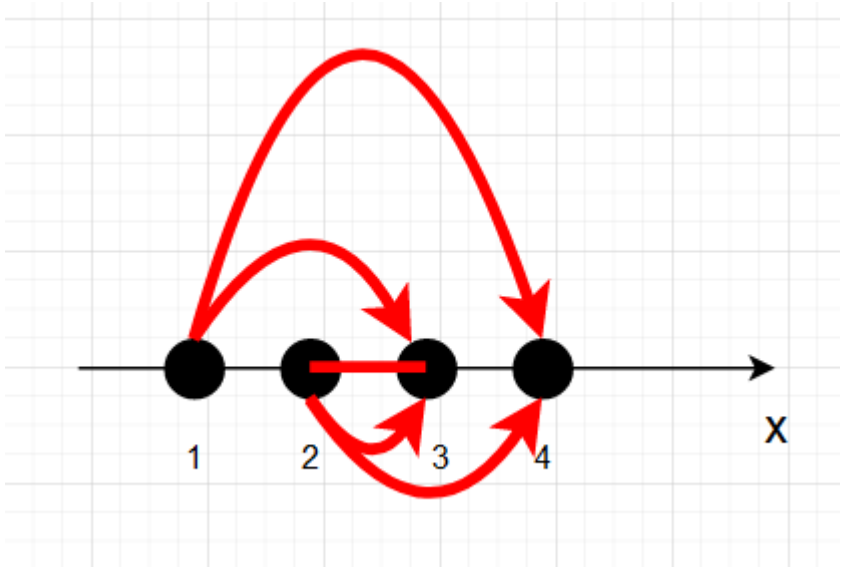


- How many time path between points 2 and 3 will be covered?



Problem #1726 "Visits"

- From every point to the left of the line to every point to the right
- In this case – 2 points to the left and 2 points to the right => 2 * 2 = 4 paths



- And vice versa in reverse direction => can be skipped

Problem #1726 "Visits"

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

- The formula for sum of path's parts parallel to axis Ox:

$$S_x = \sum_{i=1}^{n-1} (x_{i+1} - x_i) * i(n-i)$$

- Here i – number of points to the left of line between neighboring points i and i +1
- No need in abs, because all points are *sorted*
- The same for moving parallel axis Oy

Problem #1726 "Visits"

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

- Final formula:

$$Avg = \frac{S_x + S_y}{P}$$

$$= \frac{\sum_{i=1}^{n-1}(x_{i+1} - x_i) * i(n-i) + \sum_{i=1}^{n-1}(y_{i+1} - y_i) * i(n-i)}{P} =$$

$$\frac{2}{n(n-1)} * \sum_{i=1}^{n-1}(x_{i+1} - x_i + y_{i+1} - y_i) * i(n-i)$$

- For sorted arrays of x and y coordinates can be calculated for linear time
- Complexity is $O(n \log n)$, because of sorting

Problem #1521 "War Games 2"

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=1521&locale=en

- In accordance with the scheme, the war games are divided into N phases; and N soldiers, successively numbered from 1 to N, are marching round a circle one after another. At each phase, a single soldier leaves the circle, while the others continue to march. At some phase, the circle is left by a soldier, who is marching K positions before the one, who left the circle at the previous phase. A soldier, whose number is K, leaves the circle at the first phase.
- You should output the numbers of soldiers as they leave the circle.

- **Important note.** It is expected that complexity of student's solution will be not worse than $O(n \log n)$.
  Quadratic complexity isn't acceptable.

Problem #1521 "War Games 2"

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

- Naive algorithm
- Let's iterate over linked list (or array) to find k^{th} soldier each time
- In example – 6 soldiers and find 4^{th}
- Possible complexity $\sim O(k^n)$

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

Problem #1521 "War Games 2"

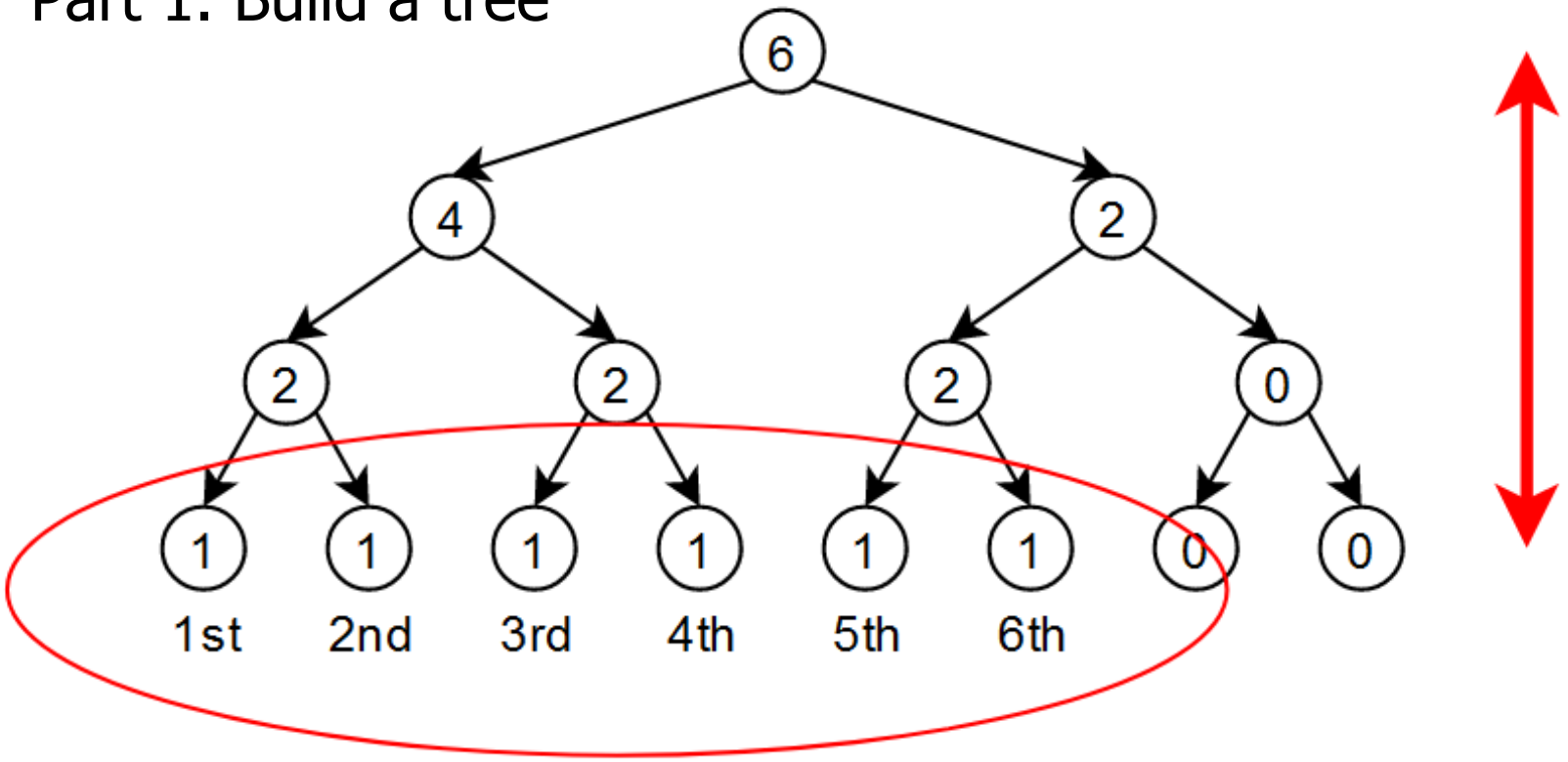
HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Similar to finding k^{th} order statistic
- Fast search, but also fast updates
- Segment tree
  - How to build it?

Problem #1521 "War Games 2"

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

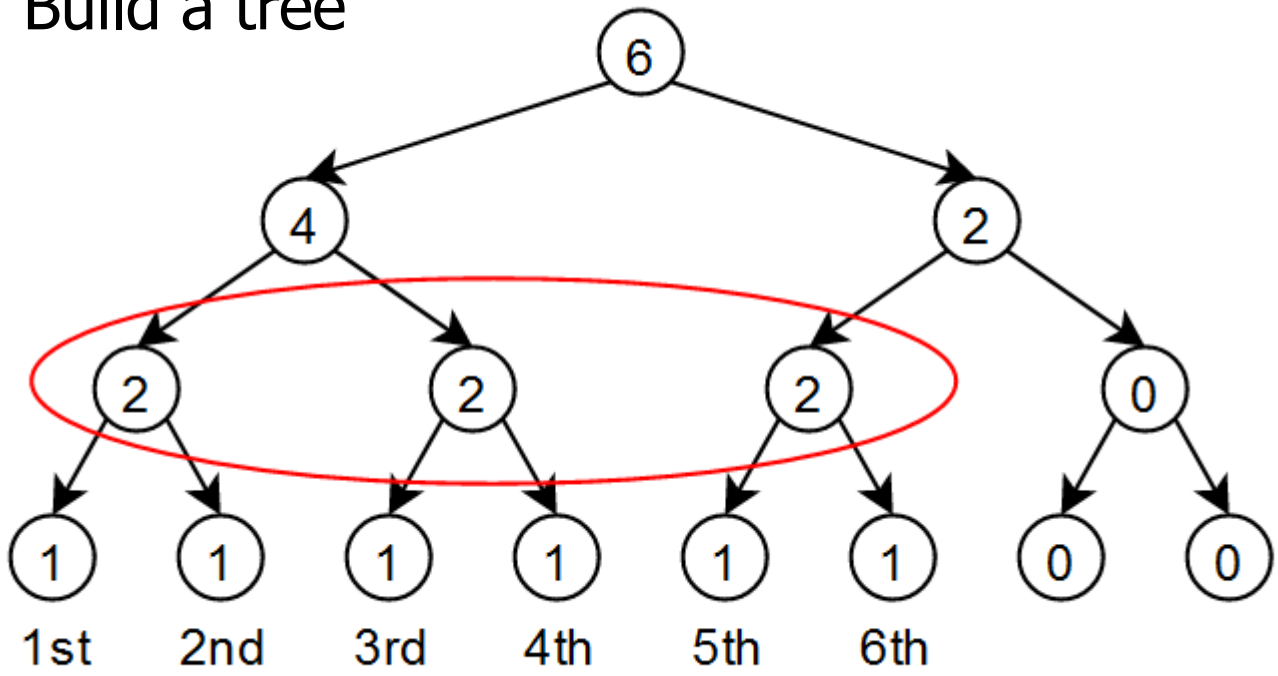
- Part 1. Build a tree



- Leaves are soldiers, each has value equal to 1
- Number of leaves is $2^{h-1} < N \leq 2^h = Q$, and height is h +1
- In example – $2^2 < 6 \leq 2^3 \Rightarrow Q = 8,\ h = 4$

Problem #1521 “War Games 2”

- Part 1. Build a tree

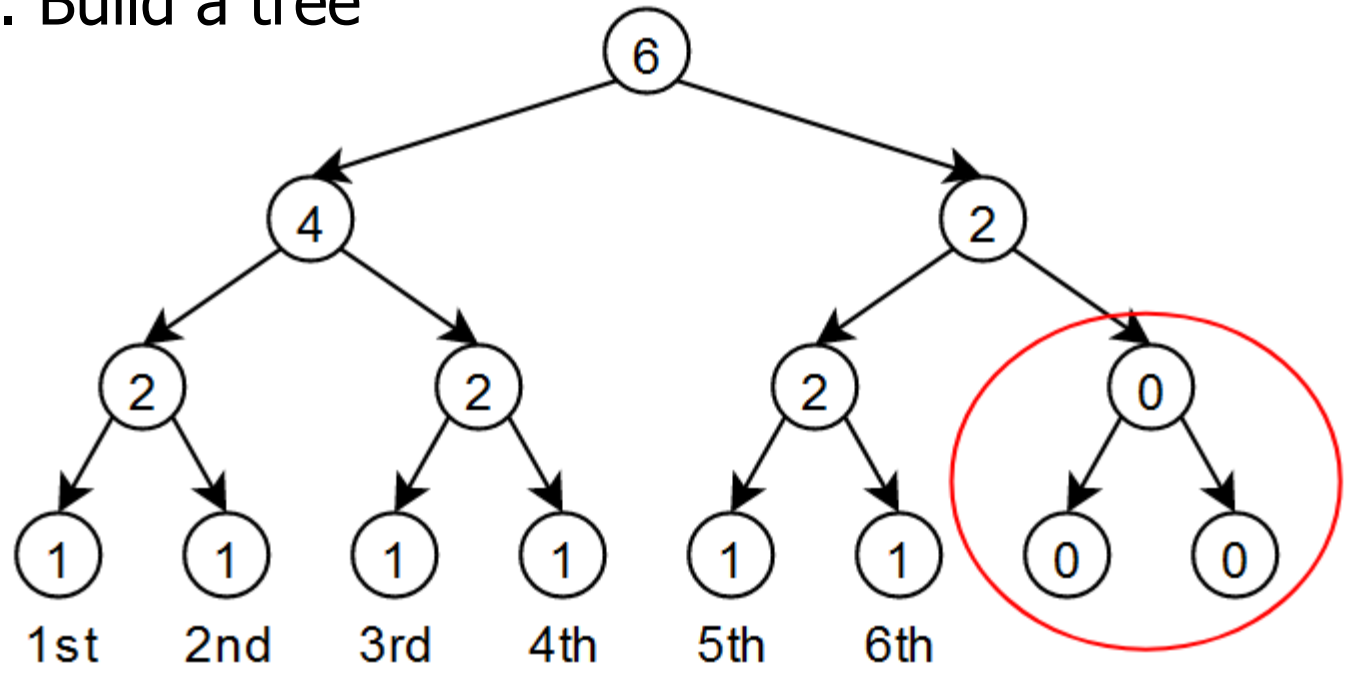


- Value in parent nodes are sum of values of its children's nodes

Problem #1521 “War Games 2”

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

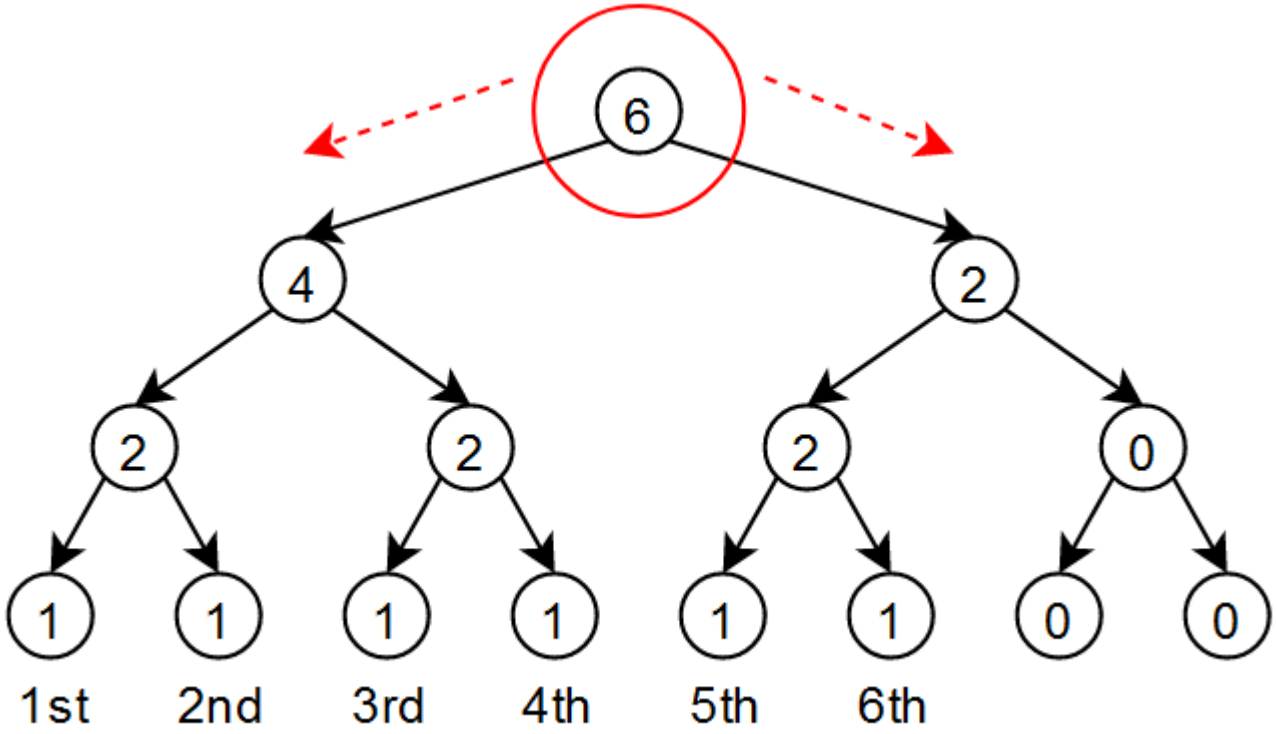
- Part 1. Build a tree



- Fill unused nodes with zeros
- Mark each leaf node with sequential number of soldier’s original position

Problem #1521 "War Games 2"

- Part 2. Search k^{th} element

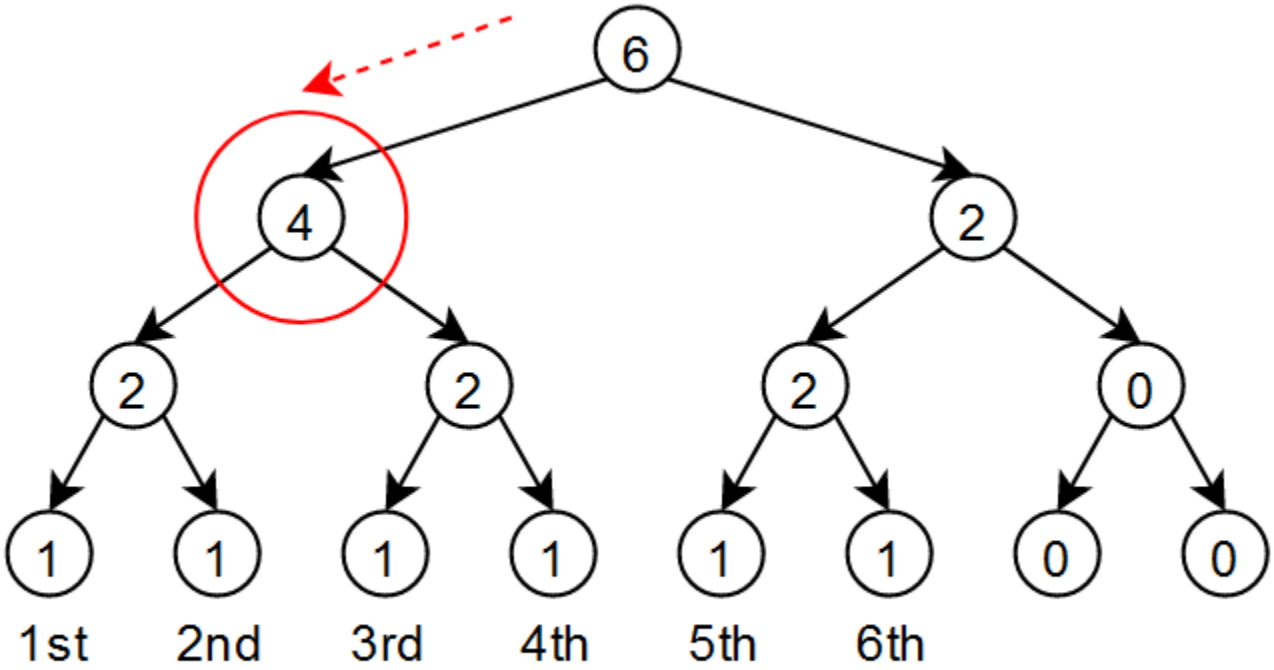


- Let's find 4^{th} element
- Start from root, need to check, should we go to the left, or to the right

Problem #1521 “War Games 2”

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

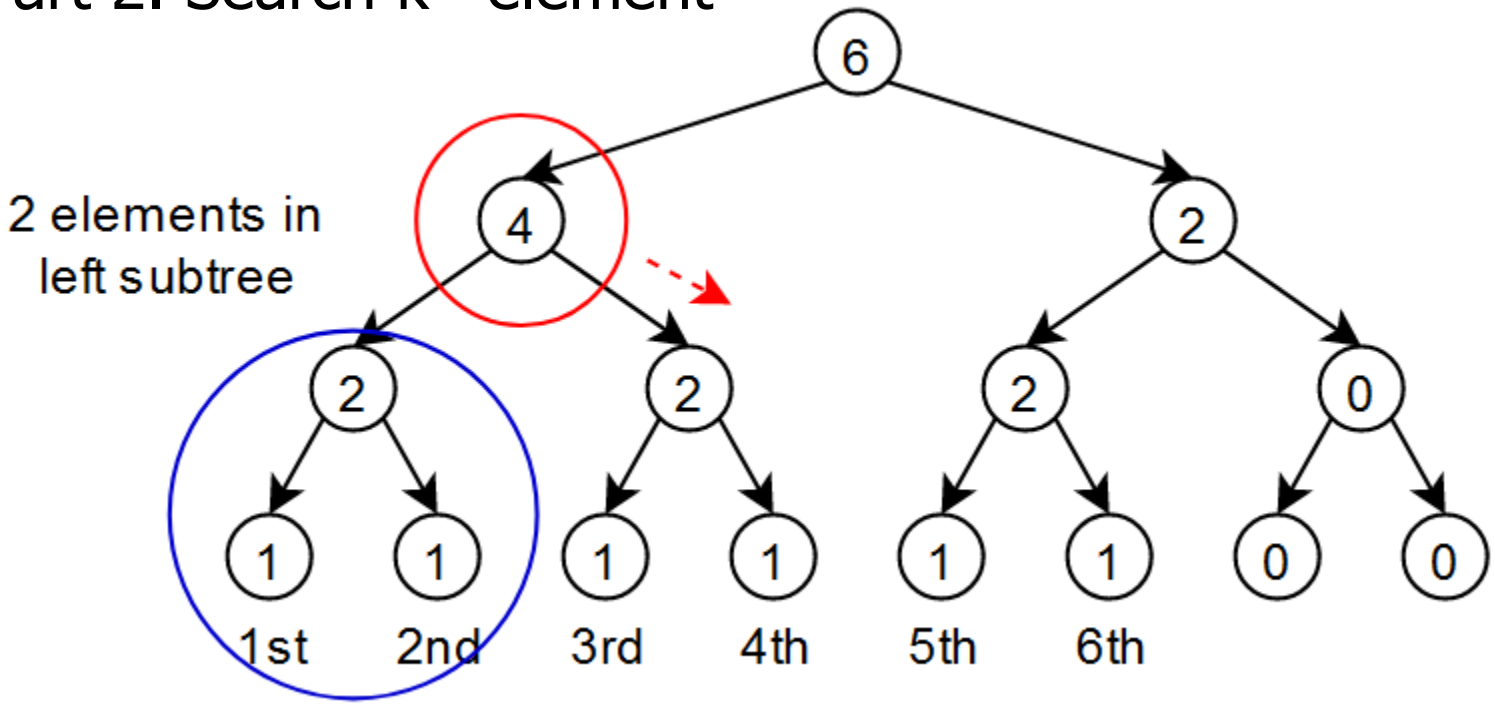
- Part 2. Search k^{th} element



- Check left child
- Is it greater or equal to searched element?
- 4^{th} element is in the left subtree
- Segment of this subtree contains elements from 1 to 4

Problem #1521 "War Games 2"

- Part 2. Search k^{th} element

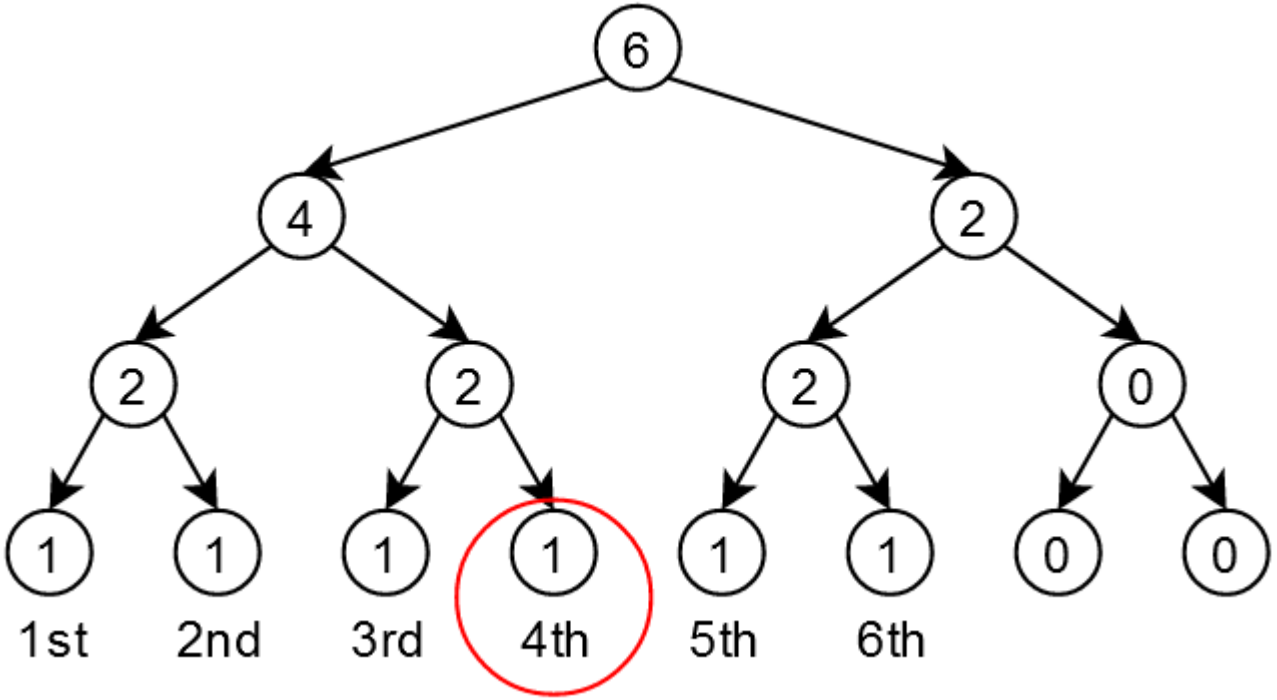


- Is left child greater or equal to searched element?
- 4^{th} element is in the right subtree
- Segment of this subtree contains elements from 2 to 4
- Modify searched element (4 – 2 = 2)

Problem #1521 “War Games 2”

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

- Part 2. Search k^{th} element

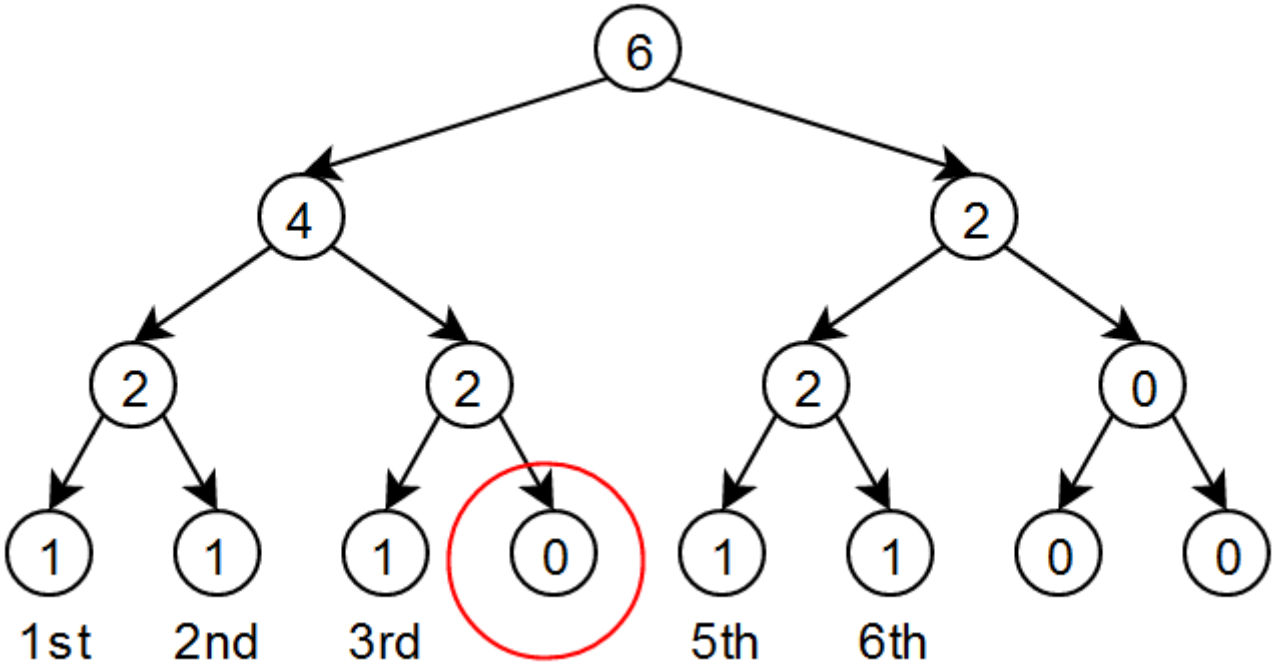


- Element is found!
- The first soldier to remove is 4^{th}

Problem #1521 “War Games 2”

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

- Part 3. Recalculate tree nodes

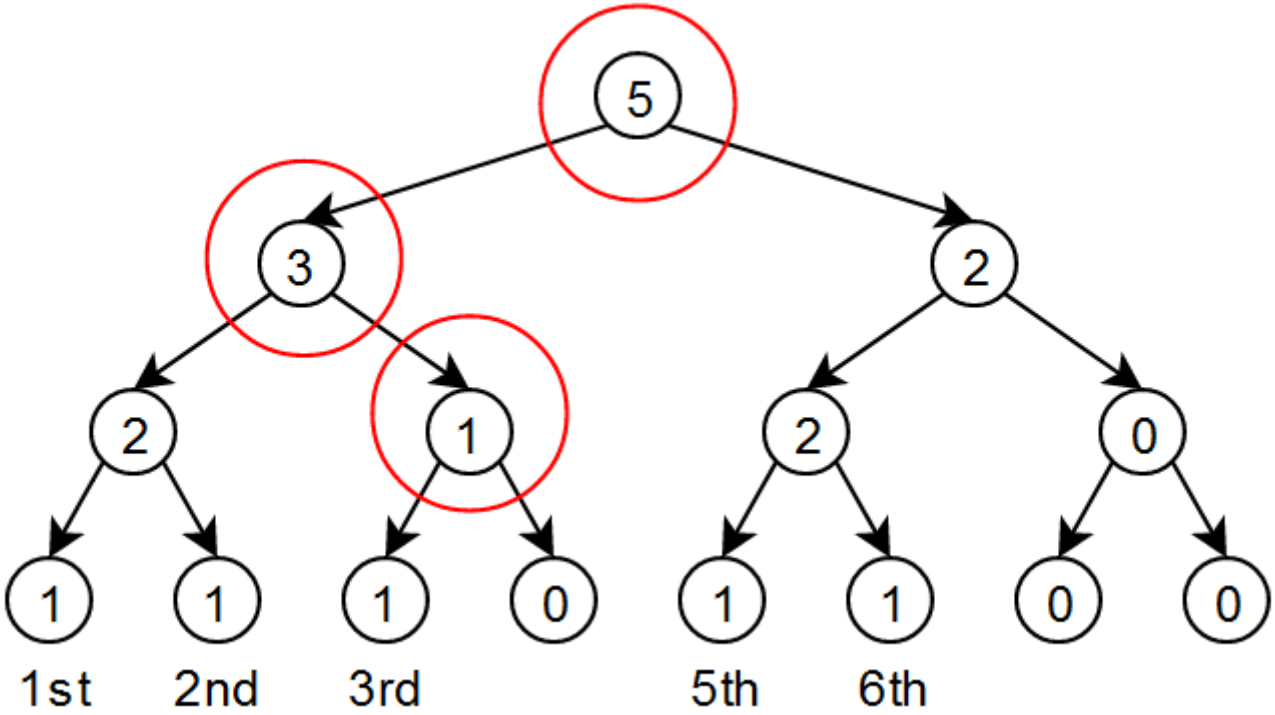


- Change value of found leaf to zero (soldier is removed)

Problem #1521 “War Games 2”

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

- Part 3. Recalculate tree nodes



- Decrease value of its parent node and so on till root node
- Now again all parent nodes contain sum of children’s values

Problem #1521 "War Games 2"

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

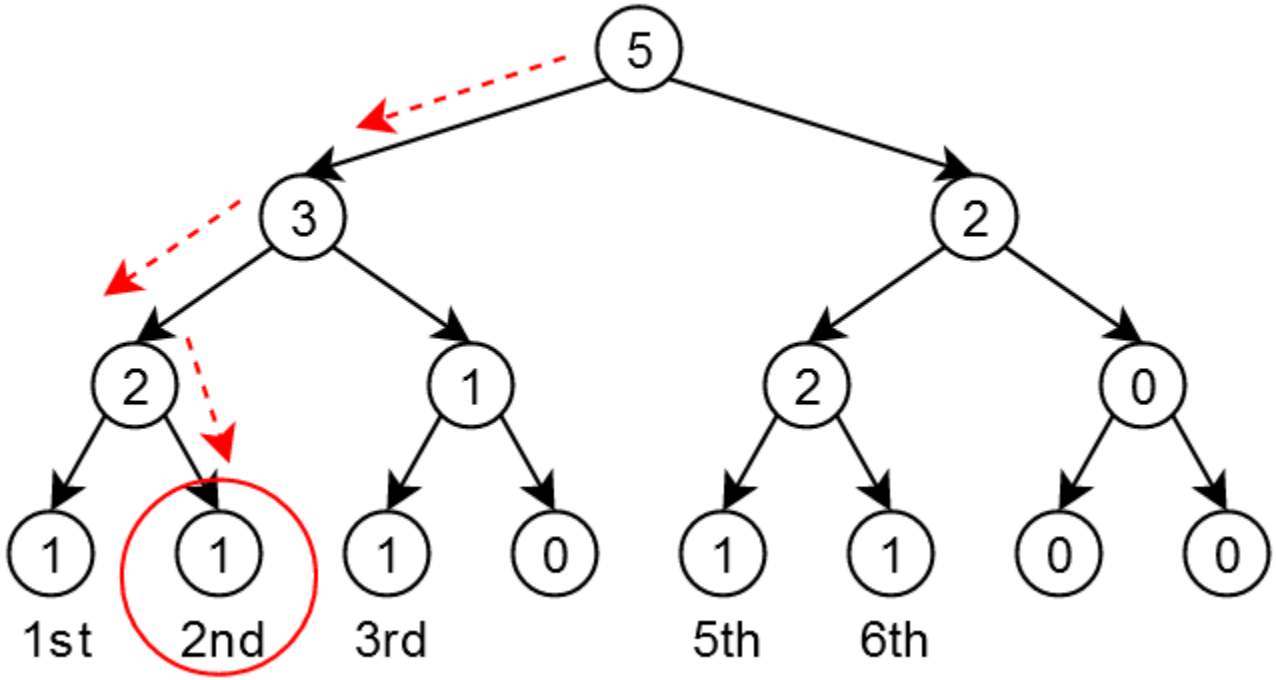
- Which node is next k[th] element?

$$k_{next} = \left(k_{found} - 1 + k\right) \% \, root$$

  - k – original k value
  - $root$ – recalculated root value
  - k_{found} – previous k_{next}
  - In example: $k_{next} = (4 - 1 + 4) \% \, 5 = 2$

Problem #1521 "War Games 2"

- Repeat part 2 and 3. Search k^{th} element and recalculate tree

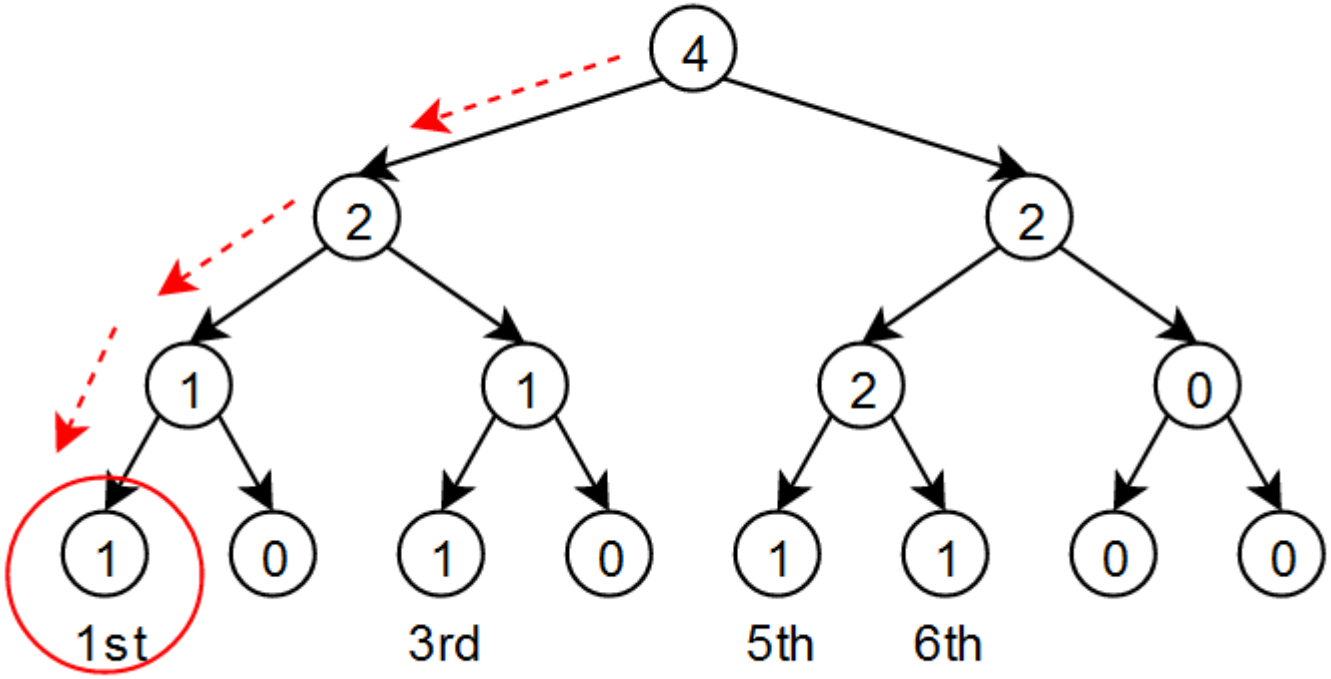


- Element is found!
- The next soldier to remove is 2nd
- $k_{next} = (2 - 1 + 4)\ \%\ 4 = 1$

Problem #1521 “War Games 2”

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

- Repeat part 2 and 3. Search k^{th} element and recalculate tree

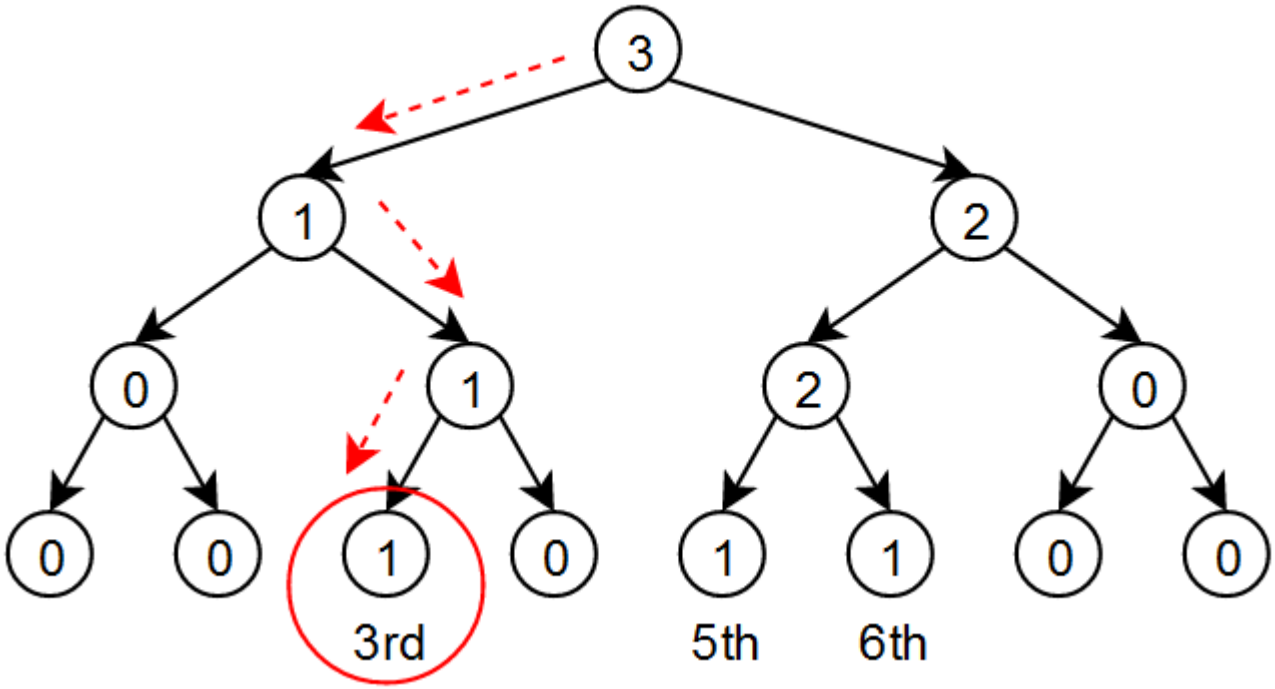


- Element is found!
- The next soldier to remove is 1^{st}
- $k_{next} = (1 - 1 + 4)\ \%\ 3 = 1$

Problem #1521 "War Games 2"

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

- Repeat part 2 and 3. Search k[th] element and recalculate tree



- Element is found!
- The next soldier to remove is 3rd
- $k_{next} = (1 - 1 + 4)\ \%\ 2 = 2$

Problem #1521 "War Games 2"

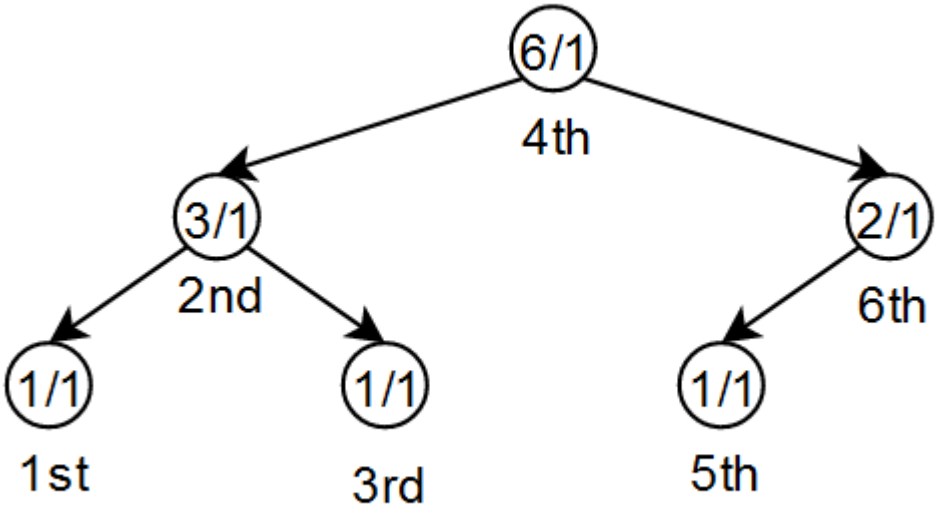
- Algorithm stops, when value of root node becomes zero
- Complexity of searching each element is logarithmic

- Advantages:
  - Can be built on array (like heap)
- Disadvantages
  - Takes a lot of extra memory for nodes

Problem #1521 "War Games 2"

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

- Binary search tree



3 – elements in subtree
1 – this node has an element

- Nodes contain number of elements in subtree and state
- Advantage: takes less memory than segment tree
- Searching can be applied in the same way

Problem #1494 “Monobilliards”

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=1494&locale=en

- The rules of monobilliards are very simple. One has to pocket successively the balls with sequential numbers into the only pocket. During the game inspector several times came up to the table and took out from the table's pocket the last of the pocketed balls. In the end it turned out that Chichikov had pocketed all the balls and the inspector had taken out and inspected them. The owner understood that this was his chance, because the inspector had to remember the order in which he had taken out the balls.
- Output the word "Cheater" if Chichikov could not pocket all the N balls in the right order, otherwise output "Not a proof".

Problem #1494 “Monobilliards”

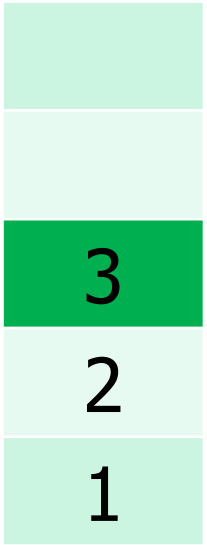
HDU-ITMO Joint Institute
杭州电子科技大学 圣光机联合学院

- Pocket is working as LIFO (last in – first out)
- Stack
- Simulation of pocket
  - Insufficient data
  - What cases are possible each time, when inspector takes a ball?

Problem #1494 “Monobilliards”

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

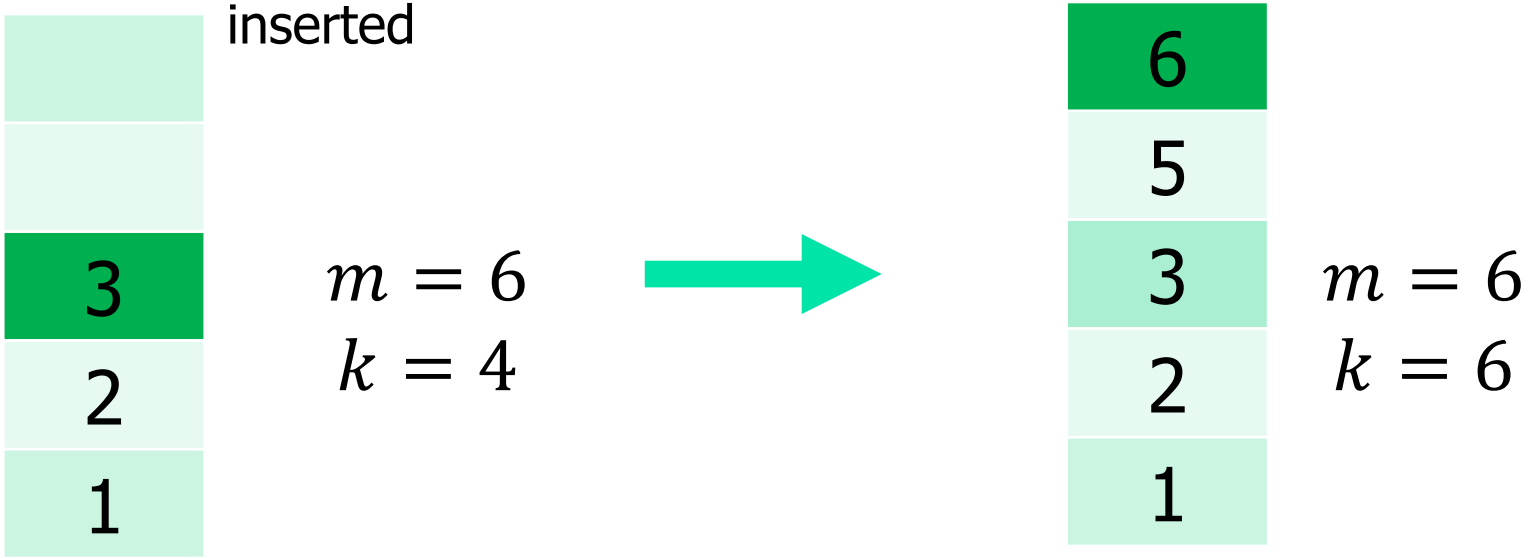
- Inspector takes a ball with number m
- Case 1. Top of stack is equal to m
  - Inspector takes expected ball from the pocke
  - It is possible, remove ball from stack and continue



$m = 3$

Problem #1494 “Monobilliards”

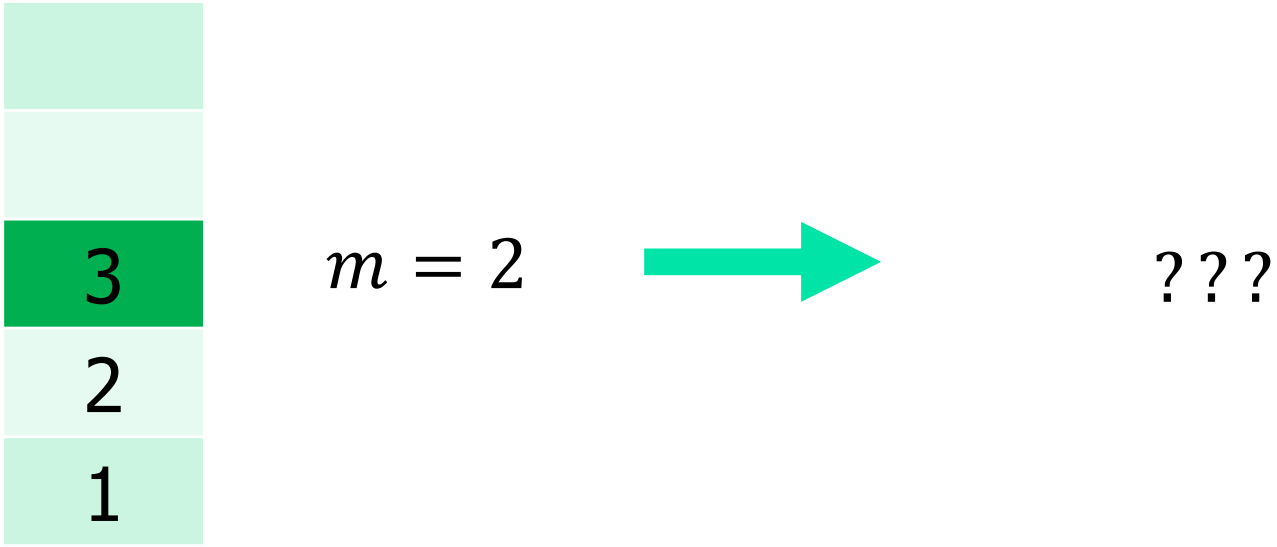
- Inspector takes a ball with number m
- Case 2. Top of stack is less than m
  - There were several balls pocketed after previous action of inspector
  - Add balls to stack in increasing order till reach m to simulate it
  - You should put only balls, which greater, than previously inserted



Problem #1494 “Monobilliards”

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

- Inspector takes a ball with number m
- Case 3. Top of stack is greater than m
  - It shouldn’t happen, because we always know the less possible top ball in the pocket
  - Player is cheater



Problem #1494 “Monobilliards”

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

- If we don’t find that player is a cheater – it isn’t a proof
- At the start you can always add ball with number 1 to the pocket (stack)

Mandatory task

1. Prepare source code to solve problem #1521 “War Games 2” https://acm.timus.ru/problem.aspx?space=1&num=1521&locale=en
2. Pass tests on Timus system for this problem https://acm.timus.ru/submit.aspx?space=1&num=1521
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 problems to get extra 2 points for each problem:

1. Problem #1494 “Monobilliards”
   https://acm.timus.ru/problem.aspx?space=1&num=1604&locale=en
   Solution of this problem was already explained
2. Problem #1067 “Disk Tree”
   https://acm.timus.ru/problem.aspx?space=1&num=1067&locale=en
   Report for this problem should contain explanation of used data structures

Thank you!