Seminar 1: Asymptotic Complexity

Solved examples

Tomáš Chamra   Marko Genyk-Berezovskyj

Faculty of Electrical Engineering
Czech Technical University in Prague

Algorithms
Spring 2017

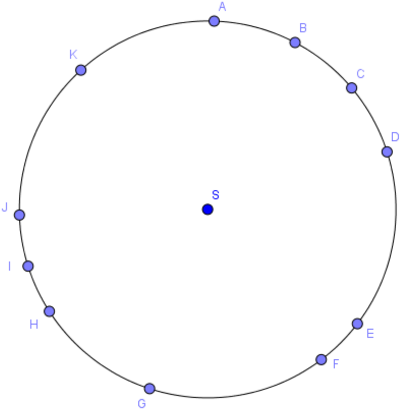
Triangles on a Circle

Assignment as proposed by Tomáš Valla

- There are N points labeled 1, 2, ..., N which are irregularly positioned on the perimeter of a given circle. The task is to compute the number of such triangles which vertices are in the labeled points and which do not contain in their interior the circle center.
- Suggest an algorithm and determine its asymptotic complexity.
- Solve an analogous problem with convex quadrilaterals instead of triangles.

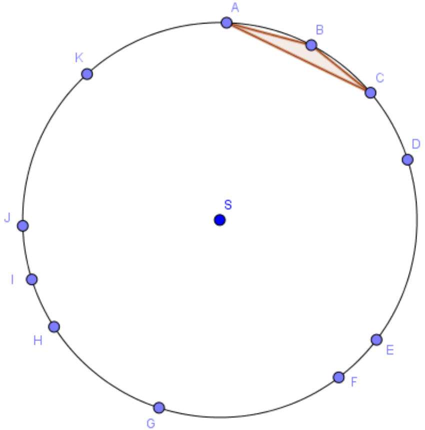
Triangles on a Circle

Solution I



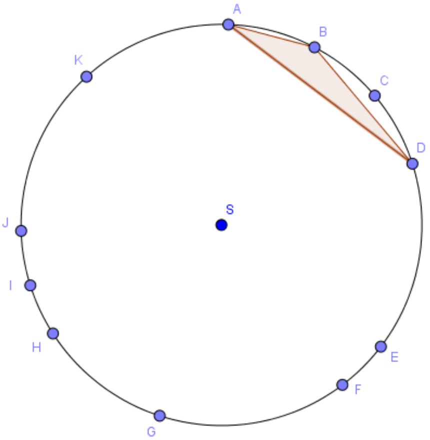
Triangles on a Circle

Solution II



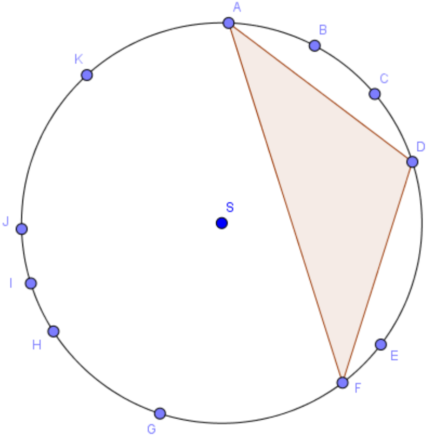
Triangles on a Circle

Solution III



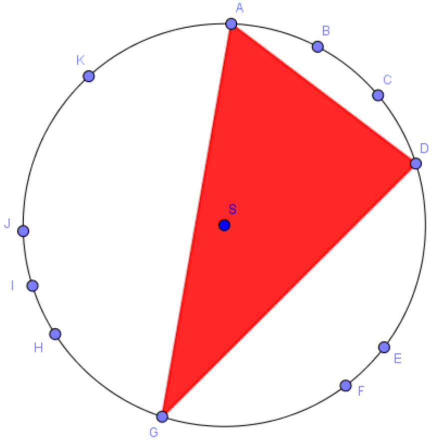
Triangles on a Circle

Solution IV



Triangles on a Circle

Solution V



Triangles on a Circle

Solution VI

- All triangles have less than 180° between first and last point

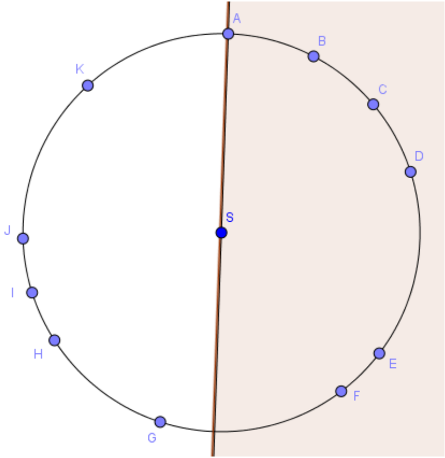
Triangles on a Circle

Solution VI

- All triangles have less than 180° between first and last point
- Idea: Sliding window
  - Window covering 180° range (excluding exactly 180°)
  - Always starts in some point on circle
  - We can make triangles consisting of starting point and any other two points in the window
- Works exactly the same way for convex quadrilaterals, with the same complexity

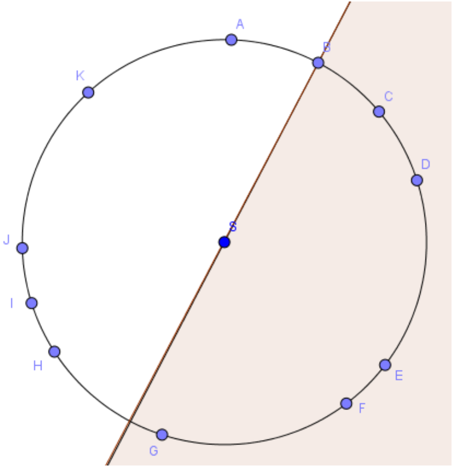
Triangles on a Circle

Solution VII



Triangles on a Circle

Solution VIII



Triangles on a Circle

Solution IX

- Low complexity: $\Theta(N)$ as we only care about number of triangles but not their exact coordinates
- If we use a queue, we can monitor exact number of points present in the window
- Number of triangles for each window step:

$$\frac{(P-1)\cdot(P-2)}{2}$$

where P is number of points in the window, including the starting point

Binary Representation

Assignment

- The task is to print all such positive integer smaller than N which binary representation contains exactly three 1's.
- What is the asymptotic complexity of an effective algorithm?
- We do not consider an algorithm linear in N to be effective.

Binary Representation

Solution I

- For illustration purposes, we have the number 485
- Binary representation of this number is 111100101

Binary Representation

Solution I

- For illustration purposes, we have the number 485
- Binary representation of this number is 111100101
- We can prove that length of binary representation is equal to $\lfloor \log_2(N) \rfloor + 1$ for any positive number N
- In our example, the length is $\lfloor \log_2(485) \rfloor + 1 = 9$

Binary Representation

Solution II

- Let's find all numbers with binary representation with 9 bits, having exactly three ones in their representation
- for (0 <= i < 9):
  for (0 <= j < i):
  for (0 <= k < j):
  number = (1 « i) | (1 « j) | (1 « k)
- Finall, we have to check the results if they're ≤ 485

Binary Representation

Solution III

- Complexity is $\Theta(log(N) \cdot log(N) \cdot log(N)) = \Theta(log^3(N))$

Binary Representation

Solution III

- Complexity is $\Theta(log(N) \cdot log(N) \cdot log(N)) = \Theta(log^3(N))$
- By the way, $log^p(N) \in O(N)$ for any positive p

Crazy Factorial

Assignment

- Describe how to find the value for $N = 10^7$:

$$\log_{10}\left(\log_{10}\left(N^{(N!)}\right)\right)$$

- How long will it take to your personal computer to compute the value? The base of logarithm is 10.
- Do not use approximations like Stirling's formula etc.

Crazy Factorial

Solution I

- We have our original formula:

$$\log_{10}\left(\log_{10}\left(N^{(N!)}\right)\right)$$

Crazy Factorial

Solution I

- We have our original formula:

$$\log_{10}\left(\log_{10}\left(N^{(N!)}\right)\right)$$

- Let's factor our the power from the inner factorial:

$$\log_{10}\left(N! \cdot \log_{10}\left(N\right)\right)$$

Crazy Factorial

Solution I

- We have our original formula:

$$\log_{10}\left(\log_{10}\left(N^{(N!)}\right)\right)$$

- Let's factor our the power from the inner factorial:

$$\log_{10}\left(N! \cdot \log_{10}\left(N\right)\right)$$

- As we know, logarithm of a product is equal to the sum of the logarithms:

$$\log_{10}\left(N!\right) + \log_{10}\left(\log_{10}\left(N\right)\right)$$

Crazy Factorial

Solution II

- By the way, factorial of $N!$ is multiplication $1 \cdot 2 \cdot \ldots \cdot N$, right?:

$$\log_{10}(\prod_{i=1}^{N} i) + \log_{10}(\log_{10}(N))$$

Crazy Factorial

Solution II

- By the way, factorial of $N!$ is multiplication $1 \cdot 2 \cdot \ldots \cdot N$, right?:

$$\log_{10}\left(\prod_{i=1}^{N} i\right) + \log_{10}\left(\log_{10}(N)\right)$$

- We still know that rule about logarithm of product...:

$$\sum_{i=1}^{N} \log_{10}(i) + \log_{10}\left(\log_{10}(N)\right)$$

Crazy Factorial

Solution III

- The assignment says that $N = 10^7$:

$$\sum_{i=1}^{10^7} \log_{10}(i) + \log_{10}(\log_{10}(10^7))$$

Crazy Factorial

Solution III

- The assignment says that $N = 10^7$:

$$\sum_{i=1}^{10^7} \log_{10}(i) + \log_{10}(\log_{10}(10^7))$$

- We need to calculate $10^7 + 2$ logarithms and 10^7 sums

Crazy Factorial

Solution III

- The assignment says that $N = 10^7$:

$$\sum_{i=1}^{10^7} \log_{10}(i) + \log_{10}(\log_{10}(10^7))$$

- We need to calculate $10^7 + 2$ logarithms and 10^7 sums
- Single processor core in your computer can perform cca $2 \cdot 10^9$ operations per second

Further Reading

T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein
Introduction to Algorithms, Third Edition.
The MIT Press, 2009.