# Упражнения: Алгоритми и сложности

Тествайте решението в Judge: <https://judge.softuni.org/Contests/4170/11-Algorithms-Intro-and-Complexity>.

Изпратете в judge една от следните стойности, съответстващи на правилната сложност

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| постоянна | логаритмична | sqrt(n) | линейна | n\*sqrt(n) |
| квадратна | n\*log(n) | кубична | 2^n | експоненциална |

## Проверка за просто число – Изчислете сложността (Най-лошия случай)

Изчислете очакваното време за работа O(f(n)) на функцията в **най-лошия случай**:

|  |
| --- |
| static bool IsPrime(long num)  {  for (int i = 2; i < num; i++)  {  if (num % i == 0)  {  return false;  }  }  return true;  } |

## Проверка за просто число – Изчислете сложността (Най-добрия случай)

Изчислете очакваното време за работа O(f(n)) на горната C# функция в **най-добрия случай**.

## Бърза проверка за четно число – Изчислете сложността (Най-лошия случай)

Изчислете очакваното време за работа O(f(n)) в **най-лошият случай** за следната функция на C#:

|  |
| --- |
| static bool IsPrimeFast(long num)  {  int maxDivisor = (int)Math.Sqrt(num);  for (int i = 2; i <= maxDivisor; i++)  {  if (num % i == 0)  {  return false;  }  }  return true;  } |

## Първите N прости числа – Изчислете сложността

Изчислете очакваното време за работа O(f(n)) в **най-лошият случай** за следната функция на C#:

|  |
| --- |
| static IList<int> FindFirstNPrimes(int n)  {  var primes = new List<int>(n);  int p = 2;  while (primes.Count < n)  {  if (IsPrimeFast(p))  {  primes.Add(p);  }  p++;  }  return primes;  } |

## Първите N прости числа – Изчислете използването на памет

Изчислете очакваното време за работа O(f(n)) в **най- средноаритметичения случай** за следната функция на C#:

|  |
| --- |
| static IList<int> FindFirstNPrimes(int n)  {  var primes = new List<int>(n);  int p = 2;  while (primes.Count < n)  {  if (IsPrimeFast(p))  {  primes.Add(p);  }  p++;  }  return primes;  } |

## Прости числа в обхвата – Изчислете сложноста

Изчислете очакваното време O(f(n)) в най-лошия случай за следната функция на C#:

|  |
| --- |
| static IList<int> FindPrimesInRange(int start, int end)  {  var primes = new List<int>();  for (int p = start; p <= end; p++)  {  if (IsPrimeFast(p))  {  primes.Add(p);  }  }  return primes;  } |

## Сравнение на бързината на две програми

Напишете програма, която сравнява бързината на функциите IsPrime(p) и IsPrimeFast(p)

|  |  |
| --- | --- |
| var startTime = DateTime.Now;  for (int i = 0; i < 50000; i++)  {  IsPrime(i);  }  var executionTime =  DateTime.Now - startTime;  Console.WriteLine("Execution time: {0}",  executionTime); | var startTime = DateTime.Now;  for (int i = 0; i < 50000; i++)  {  IsPrimeFast(i);  }  var executionTime =  DateTime.Now - startTime;  Console.WriteLine("Execution time: {0}",  executionTime); |

Попълнете следната таблица и сравнете времето (в секунди):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | p = 1 000 | p = 10 000 | p = 50 000 | p = 100 000 | p = 1 000 000 |
| IsPrime(p) |  |  |  |  |  |
| IsPrimeFast(p) |  |  |  |  |  |

Попълнете **hangs**, ако времето е повече от една минута

Този проблем няма да има оценка от judge.