

Изработили:

Савина Вълчанова, фак. номер: 471220071

Теодора Ангелова, фак. номер: 471220002

Проверил:

По

„**Паралелна обработка на информация**“

На тема:

Паралелна програма за Brute force algorithm

*София 2023г.*

**Технически университет - София**

Факултет по приложна математика и информатика

**Курсов проект**

**Съдържание:**

1. Въведение...........................................................................................2.стр.
2. Очаквани резултати от изпълнението на проекта...........................3стр.
3. Функционалност.................................................................................4стр.
4. Имплементация...................................................................................5стр.
5. Използвана литература.......................................................................7стр.
6. **Въведение**

Целта на настоящия документ е да опише софтуерните изисквания за изпълнението на примерна обществена поръчка на тема: „Brute Force Algorithm“.

**Какво представлява алгоритъма?**

Подходът с „груба сила“ е подход, който намира всички възможни решения, за да намери задоволително решение на даден проблем. Алгоритъмът за груба сила изпробва всички възможности, докато не се намери задоволително решение.

**Такъв алгоритъм може да бъде два вида:**

* **Оптимизиране:** В този случай е намерено най-доброто решение. За да намери най-доброто решение, той може или да намери всички възможни решения, за да намери най-доброто решение, или ако стойността на най-доброто решение е известна, той спира да намира, когато се намери най-доброто решение. Например: Намиране на най-добрия път за проблема с пътуващия търговец. Тук най-добрият път означава, че пътувате до всички градове и разходите за пътуване трябва да са минимални.
* **Задоволително:** Спира намирането на решението веднага щом бъде намерено задоволителното решение. Или например намиране на пътя на пътуващия търговец, който е в рамките на 10% от оптималния.
* Често алгоритмите за груба сила изискват експоненциално време. Могат да се използват различни евристики и оптимизация:
* **Евристичен:** Основно правило, което ви помага да решите кои възможности трябва да разгледаме първо.
* **Оптимизация:** определени възможности се елиминират, без да се изследват всички.

**Следните са предимствата на:** „Brute Force Algorithm“.

* Този алгоритъм намира всички възможни решения и също така гарантира, че намира правилното решение на проблем.
* Този тип алгоритъм е приложим за широк набор от домейни.
* Използва се предимно за решаване на по-прости и малки задачи.
* Може да се счита за сравнителен показател за решаване на прост проблем и не изисква познания за определена област.

В настоящото техническо задание са описани и изискванията към проектната реализация и документация.

1. **Очаквани резултати от изпълнението на проекта**

Алгоритъмът за груба сила се състои от следните стъпки:

* Генерирайте всички възмижни решения: Първата стъпка е да генерирате всички възможни решения на проблема. Това може да се направи с помощта на различни техники като пермутация, комбинации или просто итерация през всички възможни стойности.
* Тествайте всяко решение: След като се генерират всички възможни решения, алгоритъмът тества всяко едно, за да определи дали отговаря на желаните критерии.Това включва оценка на решението спрямо набор от ограничения или критерии.
* Очакваният резултат е алгоритъмът ни да избере най-доброто решение, което отговаря на желаните критерии.

1. **Функционалност**

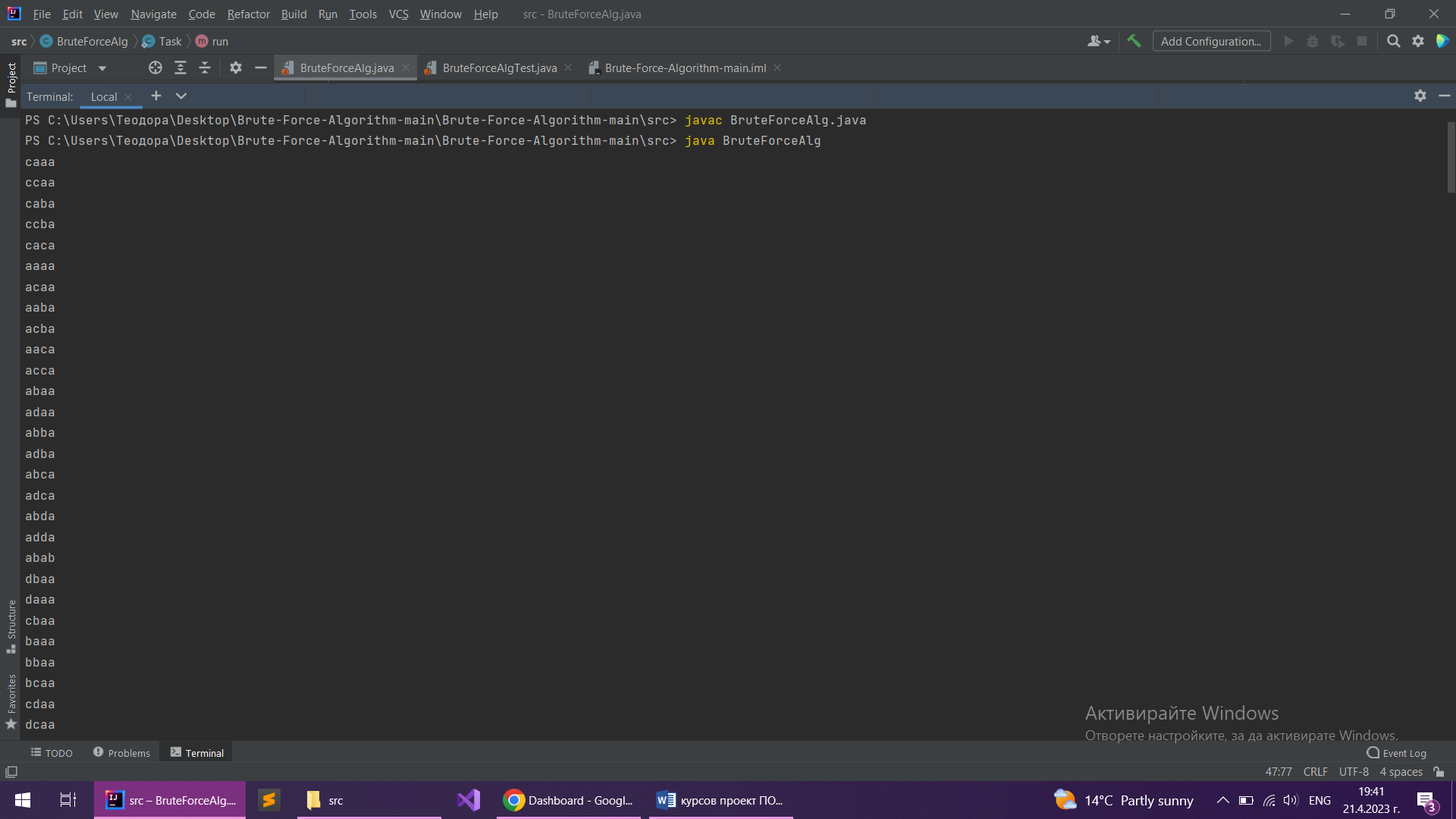
Нашият проект съдържа код на програма (с конзолен интерфейс) за брутфорс атака в Java, която търси всички възможни комбинации от символи на дължина 4 от азбука “abcd”. Разделяме задачата на няколко независими части и ги изпълваме едновременно на множество процесори.

В този пример програмата използва класа `ExecutorService` от пакета `java.util.concurrent` за изпълнение на задачите на всеки процесор. Класът `Task` e създаден за генериране на всички комбинации от символи на дължина 4 от зададена азбука, като всяка задача се изпълнява върху определен диапазон от комбинации.

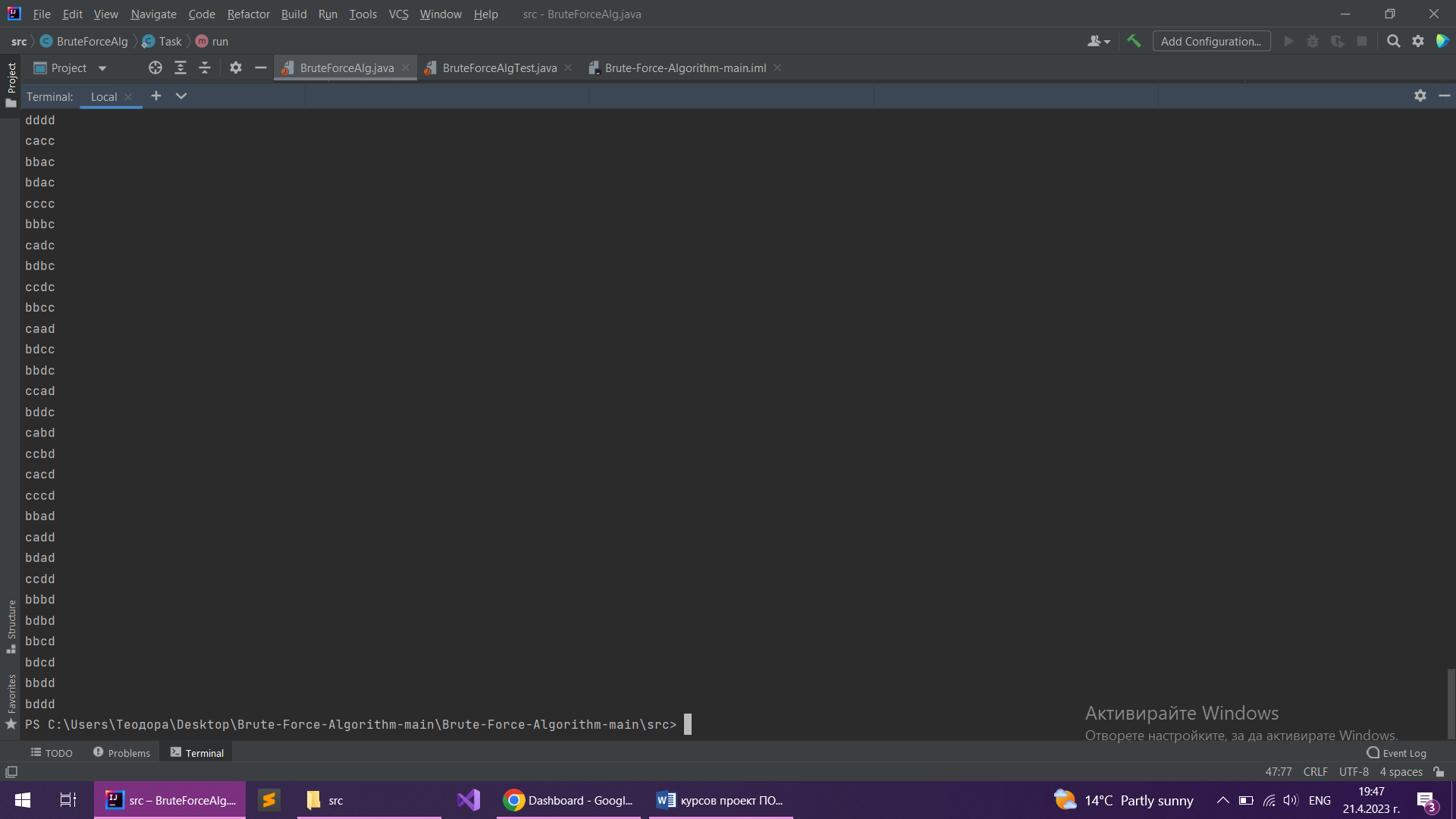
1. **Имплементация**

import java.util.concurrent.ExecutorService;  
import java.util.concurrent.Executors;  
import java.util.concurrent.TimeUnit;  
  
public class BruteForceAlg {  
 // задаване на азбуката  
 private static final char[] ALPHABET = {'a', 'b', 'c', 'd'};  
  
 // задаване на дължината на комбинациите  
 private static final int LENGTH = 4;  
  
 public BruteForceAlg(String abc) {  
 }  
  
 public static void main(String[] args) throws InterruptedException {  
 // определяне на броя процесори  
 int numThreads = Runtime.getRuntime().availableProcessors();  
  
 // създаване на ExecutorService  
 ExecutorService executorService = Executors.newFixedThreadPool(numThreads);  
  
 // изпълнение на функцията на всеки процесор  
 for (int i = 0; i < numThreads; i++) {  
 executorService.submit(new Task(i));  
 }  
  
 // изчакване на завършването на всички задачи  
 executorService.shutdown();  
 executorService.awaitTermination(Long.MAX\_VALUE, TimeUnit.NANOSECONDS);  
 }  
  
 private static class Task implements Runnable {  
 private final int start;  
  
 public Task(int start) {  
 this.start = start;  
 }  
  
 @Override  
 public void run() {  
 // намиране на всички комбинации  
 for (int i = start; i < Math.pow(ALPHABET.length, LENGTH); i += Runtime.getRuntime().availableProcessors()) {  
 StringBuilder sb = new StringBuilder();  
 int j = i;  
 for (int k = 0; k < LENGTH; k++) {  
 sb.append(ALPHABET[j % ALPHABET.length]);  
 j /= ALPHABET.length;  
 }  
 System.out.println(sb.toString());  
 }  
 }  
 }  
}

**Демо:**



От получените резултати виждаме, че нашата програма генерира всички възможни комбинации и накрая избира най-благоприятния вариант като решение, в нашия случей това е “bddd”.



След намиране на най-доброто решение, програмата приключва работата си.

1. **Използвана литература**

* <https://www.javatpoint.com/brute-force-approach>