Relatório do Trabalho Prático 1- Fundamentos de Projeto e Análise de Algoritmos

Alunos:
- Carlos Aquino
- Gabriel Ribeiro Mendonça
- Lucas Lage
- Matheus Brandão

Data: 03/11/2022

Modelagem da solução:
Tendo a problemática apresentada para nós nesse trabalho sendo a tarefa de utilizar um algoritmo no modelo de Força Bruta para resolver uma situação em que dado um conjunto de números inteiros existe um subconjunto de números dentro deste que apresenta uma soma igual a um valor estipulado nós definimos uma abordagem para soluinar essa situação.
Primeiramente, seguindo o algorimto geral para força bruta, foi necessário com que conseguiçemos listar todas as possibilidades possíveis dentro do escopo da nossa tarefa, para isso criamos uma classe responsável por retornar uma lista contendo as combinações possíveis baseadas em uma determinada sequencia.

```java
import java.util.ArrayList;
import java.util.List;

public class Combinacao {

    private static void gerarCombinacoesInterno(List<Integer> set, List<List<Integer>>
resultado, List<Integer> acumulador, int index) {
        if (index == set.size()) {
            resultado.add(new ArrayList<Integer>(acumulador));
        } else {
            acumulador.add(set.get(index));
            gerarCombinacoesInterno(set, resultado, acumulador, index + 1);
            acumulador.remove(acumulador.size() - 1);
            gerarCombinacoesInterno(set, resultado, acumulador, index + 1);
        }
    }

    public static List<List<Integer>> gerarCombinacoes(List<Integer> sequence) {
        List<List<Integer>> resultado = new ArrayList<>();
        gerarCombinacoesInterno(sequence, resultado, new ArrayList<>(), 0);
        return resultado;
    }

}
```

Em segundo lugar, continuando no fluxo desenhado do algoritmo geral para uso de força bruta, prosseguimos para a fase de avaliação das soluções geradas anteriormente. Para avaliar a soma dos sub conjuntos criamos

uma clase responsável por receber um conjunto e um valor de referencia e realizar a soma dos eles pertencentes a esse conjunto e retornar o conjunto de respostas.

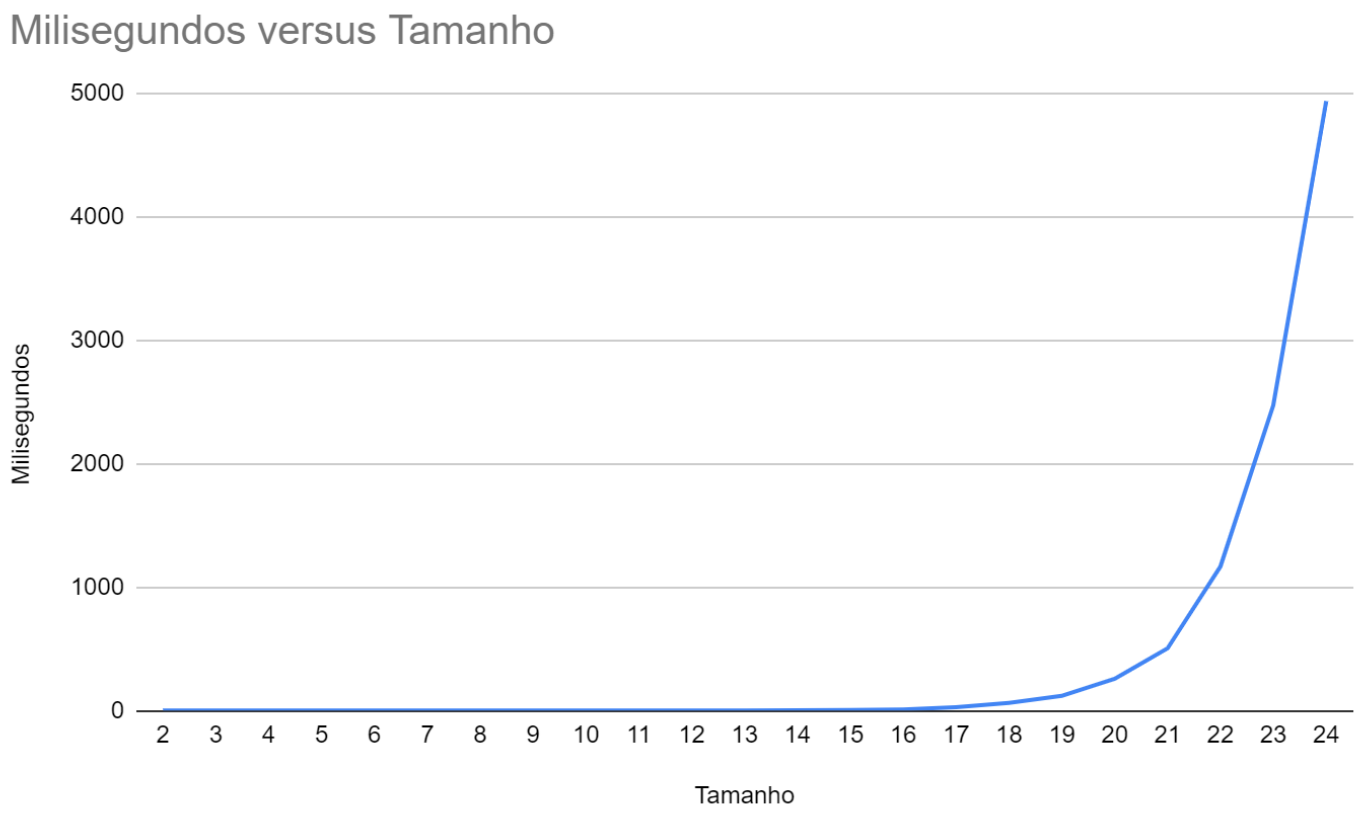
```java
import java.util.ArrayList;
import java.util.List;

public class SomaSubconjunto {

    public static List<List<Integer>> subconjuntos(List<Integer> conjunto, int soma)
{       List<List<Integer>> resultado = new ArrayList<List<Integer>>();
        List<List<Integer>> possibilidades = Combinacao.gerarCombinacoes(conjunto);
        for (List<Integer> subconjunto : possibilidades) {
            if(subconjunto.stream().mapToInt(p -> p).sum() == soma) {
                resultado.add(subconjunto);
            }
        }
        return resultado;
    }
}
```

Finalmente, nosso algoritmo deve retornar, ao final da testagem das possibilidades, o resultado encontrado. Para isso, construimos uma classe que irá realizar a execução desse método de maneira que possamos contabilizar os resultados obtidos ao realizarmos testes nesse algoritmo.

```java
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
import java.util.Random;

public class App {

    public static List<Integer> generateConjunto(int tam) {
        List<Integer> conjunto = new LinkedList<Integer>();
        Random random = new Random();
        for(int i=0; i<tam; i++) {
            conjunto.add(random.nextInt(8) + 1);
        }

        return conjunto;
    }

    public static void main(String[] args) {
        Map<Integer, Long> tempoPorTamanhoVetor = new HashMap<Integer, Long>();
        long tempoMedio = 0;
        for(int i = 2; tempoMedio <= 4000; i++) {
            List<Integer> conjunto = generateConjunto(i);
            long tempo = 0;
            for(int j = 0; j < 150; j++) {
                tempo = System.currentTimeMillis();
                SomaSubconjunto.subconjuntos(conjunto, (int)
Math.round(conjunto.stream().mapToInt(p -> p).average().orElse(0)));
                tempo = System.currentTimeMillis() - tempo;
                tempoMedio += tempo;
            }
            tempoMedio /= 150;
            tempoPorTamanhoVetor.put(i, tempoMedio);
        }
        System.out.println(tempoPorTamanhoVetor);
    }
}
```

Resultados obtidos:
Através da realização dos testes propostos conseguimos solucionar o problema em menos de 4 segundos até um conjunto de tamanho 23.

A figura abaixo mostra o gráfico de crescimento do tempo em função do tamanho do conjunto:



Pode-se observar que os conjuntos iniciais tiveram tempo próximo a 0 milissegundos visto que eram menores mas após certo ponto o tempo foi crescendo de forma extremamente acelerada.

Se o conjunto tivesse tamanho 100 o tempo em milissegundos para encontrar a soma seria de aproximadamente 1.55x10^26 milissegundos de acordo com os dados coletados.