

Cálculo de séries

Introdução

O objetivo deste exercício é calcular os seguintes problemas da forma mais rápida possível:

Algoritmo 1 - Calcular a seguinte soma

$$\sum_{i=1}^n \frac{1}{i * i}$$

Algoritmo 2 - Calcular a seguinte soma

$$S_n = e_1 \cdot e_2 + e_2 \cdot e_3 + e_3 \cdot e_4 + \cdots + e_{n-1} \cdot e_n$$

$$e_i = p_i + p_{i+1}$$

Onde p_i é o i -ésimo número primo.

Pode-se supor que o primeiro número primo é o 2. Como S pode ser muito grande, a classe `BigInteger` deve ser utilizada.

Algoritmo 3 - Calcular a constante de Brun

Ela é a soma dos inversos dos pares de primos gêmeos.

$$S = \left(\frac{1}{3} + \frac{1}{5}\right) + \left(\frac{1}{5} + \frac{1}{7}\right) + \left(\frac{1}{11} + \frac{1}{13}\right) + \left(\frac{1}{17} + \frac{1}{19}\right) + \left(\frac{1}{29} + \frac{1}{31}\right) + \cdots$$

Organização do programa

Faça um programa que aceite na linha de comando dois inteiros. Se o primeiro inteiro valer:

- 1) Calcula a soma dos n primeiros inversos de quadrado;
- 2) Calcula o valor de S para os n primeiros pares;
- 3) Calcula a constante de Brun para até o n -ésimo par de primos gêmeos.

O programa deve imprimir a resposta correta.

A classe onde está o *main* deve se chamar `Ep1`. Cada problema deve estar em uma classe diferente, e deve se chamar `Algoritmo1`, `Algoritmo2` ou `Algoritmo3`, seguindo a mesma nomenclatura do enunciado. Além disso, ele deve possuir a seguinte estrutura:

```
type calculaSoma(int n)
```

Onde *type* é o tipo com precisão arbitrária escolhida pelo usuário.

Este método recebe um inteiro n e devolve o valor pedido no enunciado.

Seu arsenal

- O EP deve ser feito em Java e pode ser feito em pares;
- **Não** é permitido utilizar bibliotecas matemáticas (ex. `Math`), os cálculos devem ser feito apenas com operações básicas.

Mais detalhes devem ser obtidos através de perguntas no Paca.

Informações gerais

- O programa deve se adaptar sem usar nenhum chamado especial à biblioteca (do tipo descobrir o número de *cores*). Para isto, pode existir uma fase de adaptação na qual o programa descobre o número de threads a serem usadas para o cálculo;
- A determinação da lista dos primos deve ser feita em tempo de execução e a utilização de tabelas de primos pré-calculados **não** é permitida;
- Os programas serão testados em 5 configurações diferentes, as duplas que obtiverem os melhores tempos em cada um dos problemas irá ganhar um ponto na média. Os bônus não são cumulativos.

Sobre a entrega

Você deve entregar um arquivo tar.gz contendo os seguintes itens:

- Arquivo(s) fonte(s);
- Makefile;
- Readme que fala como rodar e compilar a sua aplicação;
- Relatório sucinto explicando a sua solução.

O desempacotamento do seu arquivo tar.gz deverá produzir um diretório contendo esses itens. O diretório e o arquivo tar.gz deve ter nome da forma ep1-membros-da-equipe (exemplo: ep1-joaoAugusto-mariaAparecida).

A entrega deverá ser feita no paca. Se o EP for feito em dupla, **somente** uma pessoa deverá enviar o EP.

Critérios para nota

Serão avaliados os seguintes itens na composição da nota:

- Corretude;
- Uso da concorrência;
- Qualidade do código.

Importante: para cada item que não for seguido exatamente como está descrito na seção *Sobre a entrega* e seção *Organização do programa*, será descontado 1 ponto. EPs com erros de sintaxe (ou seja, erros de compilação) receberão nota ZERO.