

## Mini EP 2: A busca pelos primos que são representáveis pela soma de dois quadrados *TURBO*

Alfredo Goldman, Elisa Silva e Luciana Marques  
*MAC 0219-5742 – Programação Concorrente e Paralela 2021*

Entrega até 3 de maio de 2021

### 1. Introdução

Neste segundo mini EP buscamos otimizar os programas criados no enunciado anterior a fim de obter sua versão *TURBO*.

Vocês devem manter as linguagens de programação utilizadas na primeira versão e encontrar otimizações que aumentem o desempenho em cada uma das implementações, lembrando que linguagens diferentes podem exigir otimizações diferentes.

Apesar de *TURBO*, sua implementação deve continuar sendo sequencial (sim, a disciplina está apenas começando).

### 2. Tarefas

Otimize, sem paralelizar, os programas desenvolvidos no mini EP 1 de modo a reduzir o seu tempo de execução. É esperado que você consiga reduzir o tempo de execução em pelo menos 20%, mas existe espaço para reduções maiores.

Como o objetivo é otimizar o programa agora, vocês podem buscar algoritmos mais eficientes para encontrar primos (como o *crivo de Atkin*), mas certifiquem-se de que a solução continua correta. A seguir apresentaremos algumas dicas para melhorar o desempenho com base no pseudocódigo já apresentado:

```
01. # miniEP1
02. # Nome, N°USP
03. # comando para compilar o programa, ex: cc prog.c
04. var N = 1 << lerNumeroDaEntrada();
05. var primos = 0;
06. var primosEspeciais = 0;
07. var crivo = cria lista de Booleanos com N+1 elementos #
08. para var i = 2 até N: # inclusivo
09. >   crivo[i] = verdadeiro;
10. para var i = 2 até N: # inclusivo
11. >   se crivo[i] for verdadeiro:
12. >     primos = primos + 1;
13. >     se i % 4 != 3: primosEspeciais = primosEspeciais + 1;
14. >     para var j = i*2 até N com passo de tamanho i: # inclusivo
15. >       crivo[j] = falso;
16. imprima(primos, primosEspeciais);
```

A primeira dica é reduzir o tamanho do vetor pela metade pulando os números pares, isso permite criar um vetor com a metade do tamanho (para acessar a posição do número no vetor, basta dividir o número por 2 que se obtém o índice).

A segunda dica é usar bits para guardar os Booleanos. A ideia é que você use uma estrutura baseada em *bitarrays* para reduzir mais ainda o tamanho do vetor.

A terceira dica é buscar rotinas e bibliotecas otimizadas para aumentar o desempenho (como a função *memset* para quem está programando em C ou a biblioteca *numpy* para quem está programando em Python, são exemplos) e flags de performance do compilador (o famoso `-O3`).

**Realize 5 medidas de tempo de execução para as entradas “24” e “26” para cada programa otimizado.**

Para verificar a corretude do seu programa, para a entrada “24” você deve obter como saída “1077871 538764” e para “26” você deve obter como saída “3957809 1978494”.

Para ter uma referência, a versão não otimizada de referência em C costuma demorar 1.16 segundos para a entrada de “26” e a versão otimizada em C costuma demorar 0.24 segundos (uma redução de 79%).

### 3. Entrega

Envie os resultados obtidos com informações da sua máquina no seguinte formulário:  
<https://forms.gle/DwdSSg31YqgznnY46>.

É necessário que entre com seu email USP, bem como será possível editar sua resposta depois do envio.

Envie o código fonte dos programas em um arquivo .ZIP no eDisciplinas da matéria.  
Entrega até 3 de maio de 2021.

### 4. Critério de Avaliação

Os Mini EPs usam um critério de avaliação binária (ou 1 ou 0). Para tirar 1 envie o formulário com os resultados dos experimentos e submeta no eDisciplinas o código fonte dos seus programas.

Vale reforçar parágrafo II do artigo 23 do **Código de Ética da USP**:

Artigo 23 - É vedado aos membros do corpo docente e demais alunos da Universidade:  
[...]

II. lançar mão de meios e artifícios que possam fraudar a avaliação do desempenho, seu ou de outrem, em atividades acadêmicas, culturais, artísticas, desportivas e sociais, no âmbito da Universidade, e acobertar a eventual utilização desses meios.

Mini EPs plagiados receberão nota 0.

Se tiver dúvidas, envie uma mensagem no fórum do curso ou envie e-mails para [elisa@silva.moe](mailto:elisa@silva.moe), [lucianadacostamarques@gmail.com](mailto:lucianadacostamarques@gmail.com) ou [gold@ime.usp.br](mailto:gold@ime.usp.br) com *[miniEP2]* no assunto do e-mail. Divirta-se!