Algoritmos e Estruturas de Dados - Projeto Final
Testes à Conjetura de Goldbach com a Biblioteca GMP de C

Filipe Pires 85122
João Alegria 85048

Introdução

Este relatório abarca não só a descrição do projeto a nós proposto, da solução por nós implementada e dos resultados e conclusões obtidos, como também uma breve listagem dos maiores problemas de implementação que encontrámos e uma conclusão sobre o trabalho no geral.

Conjetura de Goldbach

Nos tempos atuais, uma grande parte das teorias e conjeturas matemáticas por provar existentes devem-se à incapacidade do ser humano de percorrer todas as possibilidades que englobam o assunto em questão. Um bom exemplo deste problema é a conjetura de Goldbach.

Esta conjetura é um dos problemas por resolver mais antigos na Teoria dos Números e na área da Matemática no geral. O que o matemático Christian Goldbach afirma é que cada número par superior a 2 pode ser expresso pela soma de dois números primos. O nome que se dá a estes números é números Goldbach. A expressão da soma de dois números primos que resultam num número par tem o nome de partição Goldbach.

Apesar do esforço considerável por parte dos matemáticos para provar a conjetura, até hoje foram apenas testados e comprovados os números inteiros inferiores a 4×10^{18} (por Oliveira e Silva, em 2012). Com estes resultados, ainda que não refutem nunca a conjetura, não é possível comprovar a ideia de Goldbach.

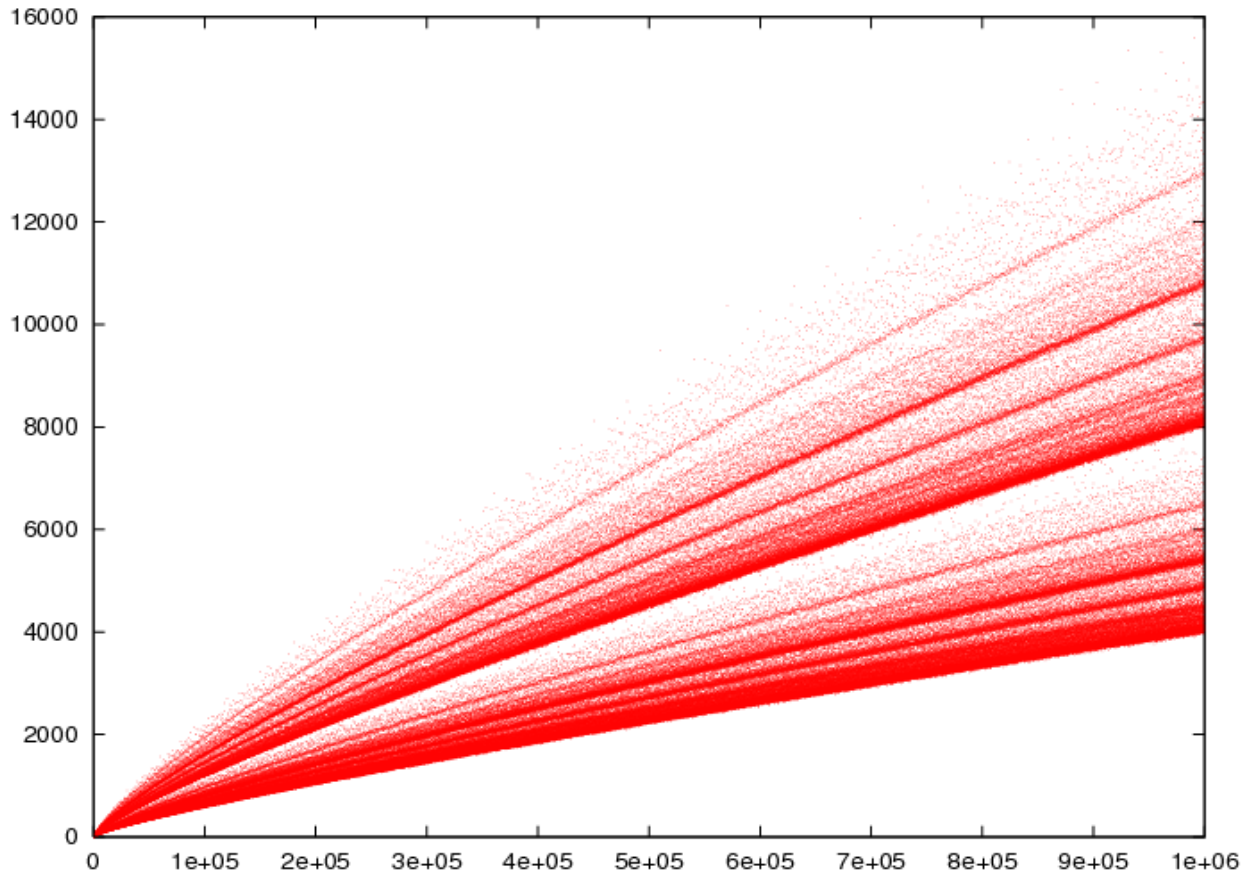

Figura 1: Número de formas de escrever 1 número par x como a soma de 2 números primos ($2 < x <= 10^6$)

Tarefa / Objetivo

A tarefa a nós proposta para este projeto resume-se ao estudo da conjetura de Goldbach e à execução de testes a esta conjetura para pares inteiros superiores a 4×10^{18} com o auxílio da linguagem de programação C e de uma biblioteca específica de nome GNU Multi-Precision Library (ou biblioteca GMP).

Esta biblioteca para precisão aritmética arbitrária permite-nos operar de forma rápida sobre inteiros cujo limite de tamanho depende apenas da memória disponível na máquina em que é corrida, dando-nos a oportunidade de testar a conjetura numa gama de tamanho considerável com valores incrivelmente elevados.

Através do conjunto de funções (com interfaces regulares) da biblioteca, tivemos como objetivo testar a conjetura de Goldbach numa gama de 10^6 (um milhão) iniciando em 4×10^{18}.

Solução / Implementação

De forma a alcançar o nosso objetivo, o primeiro passo a tomar foi o de calcular os números pares na gama de valores a ser testada, armazenados num *array* (mpz_t pares[]). Esta gama de valores teve início no número 4×10^{18} pois, como já foi anteriormente mencionado, para todos os inteiros inferiores a este, a conjetura já foi comprovada, e estendeu-se por um milhão de inteiros (número previamente definido por conveniência, ainda que possa ser alterado de forma a comprovar uma gama maior de valores). No esquema gráfico da figura 2, podemos observar esta gama de valores pela barra da direita, sendo o *m* referente ao início da gama (4×10^{18}) e o *r* referente à extensão da gama (10^6).

Sendo necessário encontrar dois números primos cuja soma dê cada um dos números pares do *array* anteriormente criado, tínhamos duas opções de implementação: ou percorreríamos todas as possibilidades de somas de números primos inferiores ao valor *m* (o que, em termos computacionais, seria praticamente impossível); ou adotávamos uma metodologia mais lógica e procurávamos as somas onde um dos operandos fosse o de menor valor possível.

Feita a escolha óbvia, prosseguimos para o cálculo e armazenamento do primeiro conjunto de números primos dos inteiros: os primeiros na gama de valores de 0 até *k* (observar figura 2), aos quais damos o nome de *little_primos* e armazenamos num *array* auxiliar (*unsigned long long* little_primos[]). Calculados os primos pequenos da operação a ser feita, restava determinar outro conjunto de primos de valores mais próximos de *m*. Este conjunto, chamaremos agora por conveniência de *big_primos*, iniciou-se em *m-k* e estendeu-se até *m+r* (constantes, estas, já definidas).

Os *big_primos* não chegam a ser armazenados em memória pois, à medida que são calculados, para cada *big_primo* é percorrido o *array* dos *little_primos* à procura da soma cujo resultado seja um dos elementos do *array* pares[]. Assim que encontrada a soma, é armazenado num *array* auxiliar inicializado a zeros (*unsigned long long* found[]) o operando *little_primo* no índex igual ao do par em questão (no *array* pares[]).

A conjetura é então comprovada se no array found[] não existir nenhum elemento de valor zero.

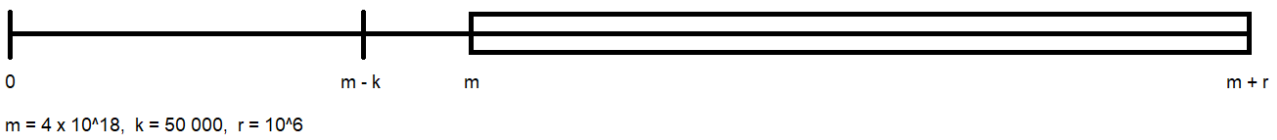

Figura 2: esquema visual da implementação adotada

Após o teste bem sucedido para uma gama de 10^6 números, decidimos ser mais ambiciosos e correr o programa para 10^8 percorrendo-a múltiplas vezes para valores de *m* dferentes e guardando num ficheiro de texto as diferenças entre 2 primos consecutivos de todas as procuras feitas pelo programa. No entanto, como a duração da execução deste segundo teste era superior ao número de dias que tínhamos disponíveis, deixamos abaixo um gráfico deste mesmo teste mas para uma gama inferior (10^5) exequível em tão curto espaço de tempo.

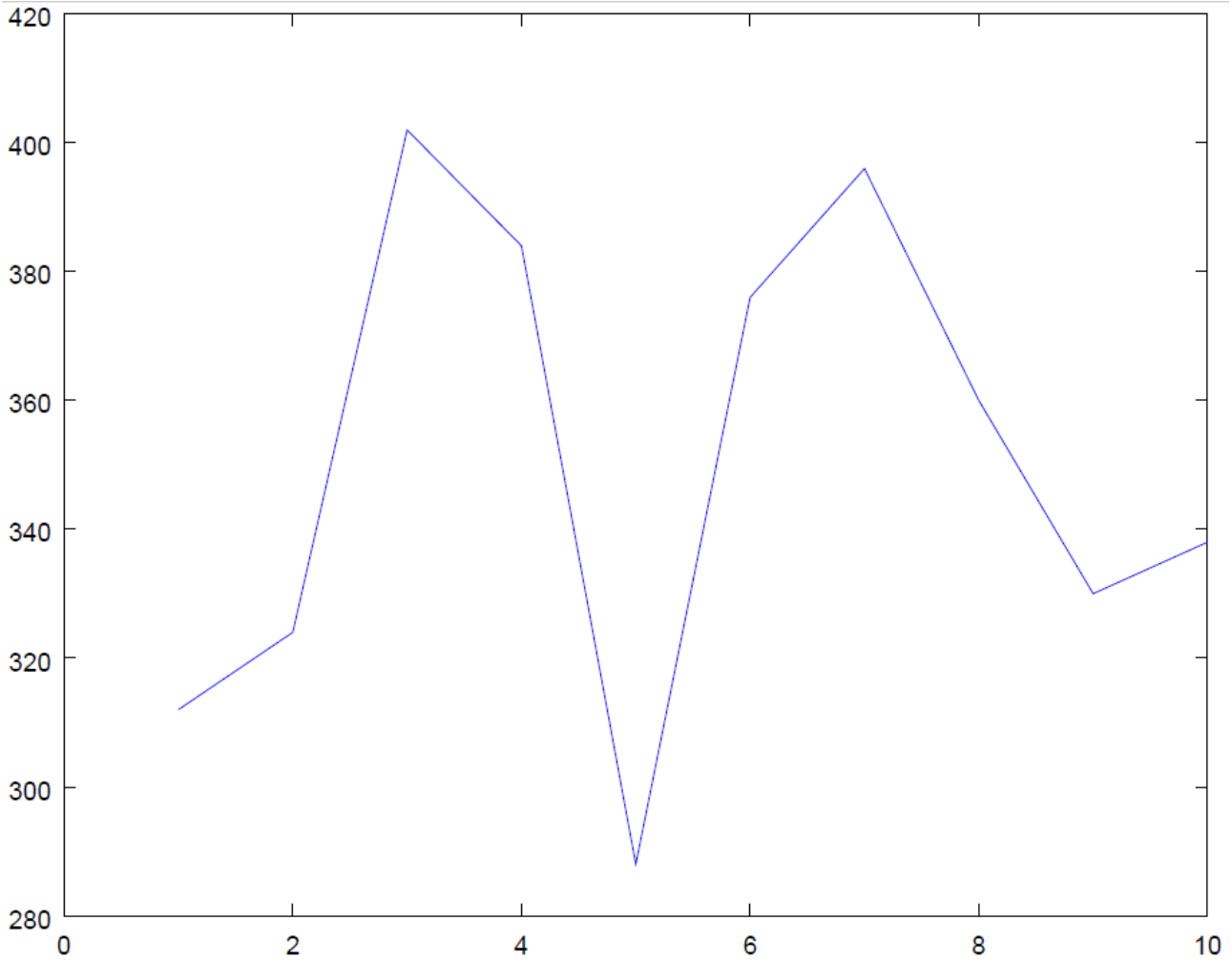

Figura 3: gráfico que relaciona os valores de *m* (onde m = $4*10^{18} + 2x*10^8$) com a diferença entre os dois maiores primos consecutivos

Problemas / Dificuldades

Na primeira fase do projeto o maior desafio a enfrentar foi claramente a adaptação à sintaxe da interface de utilização da biblioteca GMP. Após uma certa perseverança e um estudo mais aprofundado sobre as funções da biblioteca, ultrapassámos esta dificuldade sem perder muito tempo.

Já na fase da implementação em concreto, não houve problemas de gravidade relevante, sendo que apenas iam ocorrendo erros de sintaxe e de comparações desnecessárias que abrandavam o algoritmo.

A apontar relativamente ao segundo teste feito à conjetura de maior profundidade, temos apenas o facto de que os valores do gráfico obtido após uma avaliação aos números primos encontrados consecutivos não parecem mostrar uma relação notória entre os números pares para os quais é feito o teste e os números primos que solucionam o teste.

Numa visão mais geral do desenvolvimento do projeto, podemos afirmar que a implementação da solução foi feita de uma forma bastante fluída sem impasses de longa duração.

Conclusão

Com este projeto aprendemos não só a utilizar uma das bibliotecas da linguagem C, como também como comprovar uma das conjeturas mais antigas e mais famosas da teoria dos Números numa determinada gama de valores.

A nível do desempenho dos elementos do grupo, podemos concluir que trabalhamos bem como par e mantemos uma comunicação constante de forma a estarmos sempre a par do ponto de situação do trabalho, resultando num bom desempenho neste tipo de tarefas.

Quanto à solução por nós implementada, podemos concluir que é perfeitamente funcional, segundo os testes a ela feitos e pensamos que a forma como está construída tem um desempenho aceitável, visto que as únicas formas de comprovar a conjetura englobam sempre pesquisas exaustivas.

Quanto à veracidade da conjetura para a gama de valores testada, podemos concluir que não encontrámos situação nenhuma na qual exista um par cujo valor não possa ser encontrado através da soma de dois números primos, ou seja, a conjetura mantém-se verdade para todos os valores na qual foi testada.

Fontes de Pesquisa:

https://stackoverflow.com/
https://www.tutorialspoint.com/cprogramming/index.htm
https://www.tutorialspoint.com/c_standard_library/index.htm
http://mathworld.wolfram.com/GoldbachConjecture.html
https://artofproblemsolving.com/wiki/index.php?title=Goldbach_Conjecture
https://upload.wikimedia.org/wikipedia/commons/7/7c/Goldbach-1000000.png~
https://gmplib.org/
https://www.cs.colorado.edu/~srirams/courses/csci2824-spr14/gmpTutorial.html
https://gmplib.org/manual/Number-Theoretic-Functions.html#Number-Theoretic-Functions
https://gmplib.org/manual/Integer-Arithmetic.html#Integer-Arithmetic