

O projeto encontra-se dividido em dois uma vez que para a busca binária faz-se necessária a ordenação dos vetores gerados aleatoriamente. Ambos foram escritos em *\*.py* utilizando de bibliotecas conhecidas e algumas em particulares como *lmfit* para determinação da função mais próxima a ser modelada no final gráfico. A *astroML* e *time* foram para construção e design gráfico, formato *Latex*, e para terminação pela função *time.time()* do tempo de execução.

Na busca linear, não ordenada, foi verificada a possibilidade de encontrar elemento a elemento o máximo valor do vetor dentro de uma amostra (*samples*) de tamanhos (*N*) variados. A função *time* colocada na início do laço mede as interações e retorna o tempo ao final quando o valor desejado é portanto encontrado. Não se buscou salvar os valores mas apenas criou-se uma lista vazia e um novo laço para armazenar os tempos (*tmaximo*) para cada valor de *N*. Por implementação própria, sabendo do conceito de função linear, apliquei o modelo linear da biblioteca *lmfit* que ajusta curvas por minimização utilizando de erros médios quadráticos, imprimindo em tela os valores de  $\chi^2$  dentro outras estatísticas inclusive bayesiana. Verifica-se portanto a função de primeiro grau conforme a figura 1.

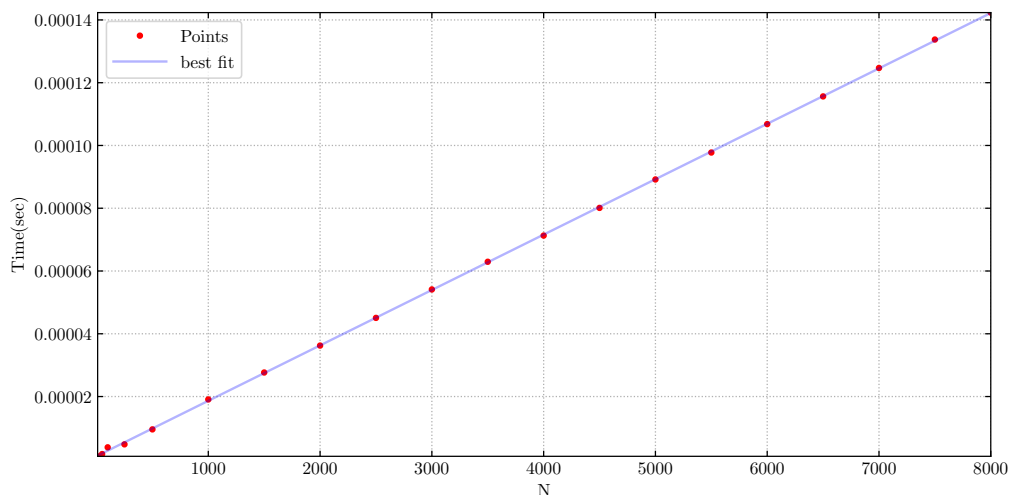


Figura 1: Busca linear

Já na busca binária, escolheu-se aleatoriamente um valor (53), como representando na figura 2, a ser encontrado de forma que em todos os valores de da amostra de novos vetores, *N*, fossem executados a busca, a priori. A amostra não foi igualmente utilizada apenas por ter sido levada em conta que utilizamos valores, na busca linear inicialmente próximos de zero a 1000 enquanto na busca binária a execução se deu inicialmente com *N* = 1000. Visando utilizar outro número, maior que o primeiro, fizemos nova busca agora com novo valor (420) e a curva característica pode ser vista na figura 3, utilizando o mesmo valor de *N*.

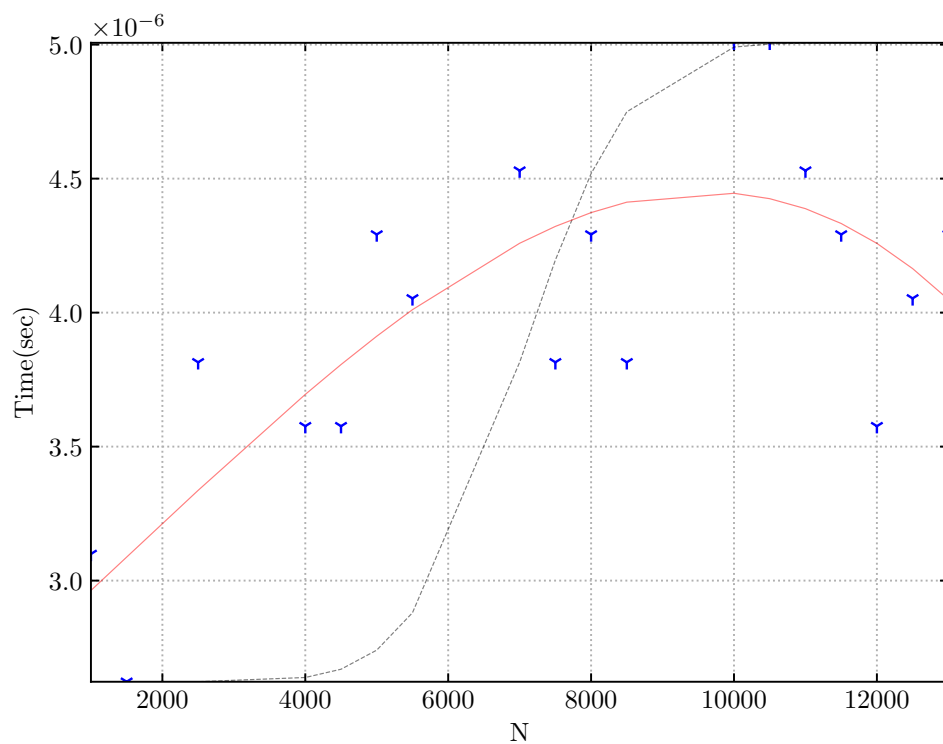


Figura 2: Primeira busca binária

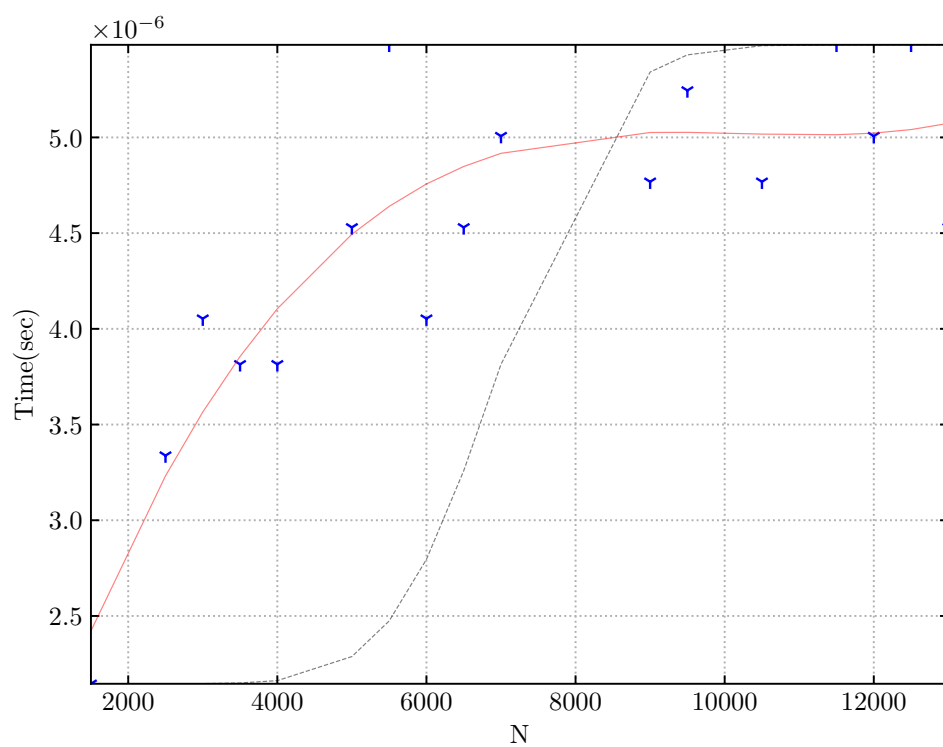


Figura 3: Segunda busca binária

## **Conclusão**

Os tempos de execução para a busca binária são, portanto menores, respeitando os padrões e eficiência de máquina. Os valores dependem da busca e da pausa executada no momento encontrado e no fato de que a busca linear, como matematicamente se refere o nome cresce monotonicamente com o número estimado de valores de uma dado vetor de dados, não sendo portanto uma busca eficaz para utilização em grande quantidade de dados.