**UNIVERSIDADE ESTADUAL DE SANTA CRUZ**

GUSTAVO BONFIM FRAGOSO

**Projeto 2 C – Quicksort**

C and Prolog implementation

Ilhéus – BA

2016

**GUSTAVO BONFIM FRAGOSO**

**Projeto 2 c – Quicksort**

C and Prolog implementation

Implementações em linguagem C e prolog que avalia o tempo de execução do quicksort para diferentes tamanhos de array.

Prof.º César Bravo

Ilhéus – BA

2016

Sumário

[1. Compilação 1](#_Toc451428737)

[2. Execução 1](#_Toc451428738)

[3. Gerando as listas de todas as combinações possíveis 1](#_Toc451428739)

[4. Conclusões 1](#_Toc451428740)

[5. Download 1](#_Toc451428741)

# Compilação

gcc -o quick quicksort.c

# Execução

Em C: ./quick N (para uma array de tamanho N)

Prolog: Abrir o arquivo correspondente ao tamanho da lista quicksortX.pl” e utilizar a regra “check”. (Ver próxima seção)

# Gerando as listas de todas as combinações possíveis

Em C o próprio código gera o arquivo de combinações que irá utilizar. Em prolog, antes de utilizar o quicksortX.pl deve-se executar o código combXgenerator.pl, onde X é o tamanho da lista, utilizando a regra “run”. (Se houver um texto de combinação pronto basta renomear para combX.txt).

# Conclusões

* O tempo de execução da implementação em C aumenta de maneira considerável entre 8 e 9;
* Os arquivos gerados para combinações 8 números, ou superiores, das implementações não foram anexados no presente exercício por conta do tamanho, (combinações de 8 números gera um arquivo de 300MB e de 9 números gera um arquivo de 12 GB);
* O programa em prolog não consegue ordenar os arquivos acima de 7, apresentando o erro de Estouro da pilha local (Out of local stack).
* O gráfico detalhado dos tempos de execução pode ser encontrado no GitHub, seção 5.

# Download

https://github.com/gbfragoso/Quicksort