

Out[30]:

The raw code for this IPython notebook is by default hidden for easier reading. To toggle on/off the raw code, click [here](#).

# Supercomputação - Projeto 1 (Entrega 1) - Rafael Vieira Rosenzvaig

## O Problema

*Este projeto tem como objetivo resolver o problema de escolha de projetos finais de engenharia dos alunos do Insper, utilizando técnicas de computação paralela.*

### Solução:

A solução encontrada foi a mais intuitiva possível. A função `escolhe_alunos()` recebe a lista de preferências de cada aluno como entrada e recursivamente calcula a satisfação geral com todas as combinações possíveis. A combinação que possuir satisfação geral melhor será a escolhida e retornada pela função.

## Infraestrutura

### Executáveis:

- `ingenua`
- `backtrack.py`
- `cria_entrada.py`

### Arquivos:

- `backtrack.py`
  - Este arquivo define as funções:
    - `escolhe_alunos()`, é a função que recursivamente calcula qual a melhor combinação possível entre a distribuição dos alunos, baseada na preferência de cada um.
    - `main()`, lê os parâmetros do arquivo de entrada e chama a função `escolhe_alunos()` com esses parâmetros.
- `ingenua.cpp`
  - Este arquivo é a tradução direta para C++ do arquivo `backtrack.py`, portanto as funções fazem as mesmas coisas.
- `cria_entrada.py`
  - Este arquivo cria uma entrada para o arquivo `backtrack.py` e/ou `ingenua.cpp` e dentro dela são definidos os parâmetros de número de alunos, número de projetos e número de escolhas que cada aluno pode fazer para ranquear sua preferência.
- `CMakeLists.txt`
  - Este arquivo contém os parâmetros de compilação para gerar o executável.

### Compilando e executando

Para compilar o programa digite o seguinte código na pasta do projeto: Primeiro `make -C ./build` e depois `cmake ./build` para usuários de Linux ou `cmake -D CMAKE_CXX_COMPILER=g++-9 -B ./build` para usuários de Mac.

Para executar o programa digite o seguinte código na pasta do projeto: `build/ingenua < entradaX` O número da entrada deve ser o da entrada desejada.

#### Resultados brutos:

Python entrada1 0.17119383811950684 segundos

C++ entrada1 0.009446144104003906 segundos

Python entrada2 0.19092702865600586 segundos

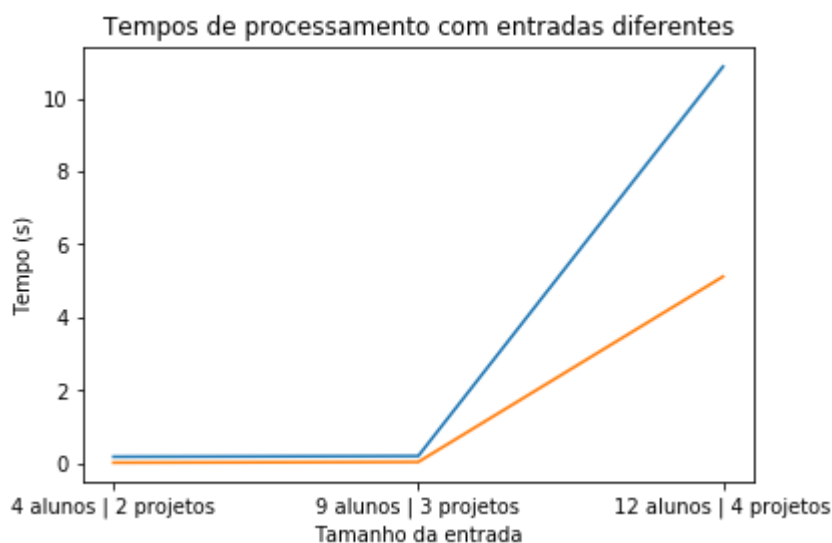
C++ entrada2 0.027827024459838867 segundos

Python entrada3 10.869203090667725 segundos

C++ entrada3 5.111987829208374 segundos

## Resultados

Observando os resultados é possível afirmar que a mudança de python para C++ proporcionou uma melhora substancial na velocidade em todos os tamanhos de entradas fornecidas: em até quase 10 vezes para entradas menores e 2 vezes para a entrada maior.



## Considerações finais