

# Relatório 2º projecto ASA 2020/2021

**Grupo:** al041

**Aluno(s):** Gonçalo Guerreiro (95581) e Vasco Correia (94188)

---

## Descrição do Problema e da Solução

Neste segundo projeto da cadeira de ASA foi-nos apresentado o seguinte problema: Dados dois processadores, um programa constituído por vários processos, os custos de execução de cada processo em cada processador e os custos de comunicação entre cada processo, qual a distribuição que minimize o custo total de execução.

Fizemos a correspondência do problema a um problema de fluxo máximo, considerando os processos e os processadores como vértices e os custos de execução e comunicação como arcos pesados num grafo dirigido, sendo o processo “X” a fonte e, o processo “Y” o sumidouro.

Para representar a rede residual utilizamos uma lista de adjacências para guardar os vizinhos de cada vértice e uma matriz que para cada posição  $[u, v]$  guarda a quantidade de fluxo que se pode enviar por esse arco.

Utilizámos o algoritmo de Edmonds-Karp (implementação do método de Ford-Fulkerson que utiliza a BFS para encontrar os caminhos de aumento) para maximizar o fluxo na rede. No final da execução do algoritmo a quantidade de fluxo que sai da fonte coincide com o custo mínimo da execução do programa.

## Referências

Cormen T., Leiserson C., Rivest R., Clifford S. - Introduction to Algorithms, *The MIT Press*

Slides disponibilizados na página da cadeira (Fluxos Máximos)

Apontamentos disponibilizados pelo professor em:

<https://web.tecnico.ulisboa.pt/jose.fragoso/asa/aula6.pdf> ;

<https://web.tecnico.ulisboa.pt/jose.fragoso/asa/aula10.pdf>

## Análise Teórica

- Leitura dos dados do “stdin”: simples leitura do input, construindo simultaneamente, a lista de adjacências e a matriz das capacidades. O custo desta etapa é dominado pela alocação e preenchimento da matriz (que depende quadraticamente do número de processos). Logo  $O(n^2)$ .
- Aplicação do algoritmo Edmonds-Karp para calcular o fluxo máximo. O algoritmo de Edmonds-Karp é uma especificação do método de Ford-Fulkerson, cuja complexidade é  $O(|f^*|E)$ , como o  $|f^*|$  é  $O(n)$  e o número máximo de arcos é  $O(n^2)$ , a complexidade total desta etapa é  $O(n^3)$ , em que “n” é o número de processos.
- Apenas é feito “print” de uma string com um número no “stdout”. Logo,  $O(s)$ , sendo “s” o comprimento da string.

Complexidade global da solução:  $O(n^3)$

# Relatório 2º projecto ASA 2020/2021

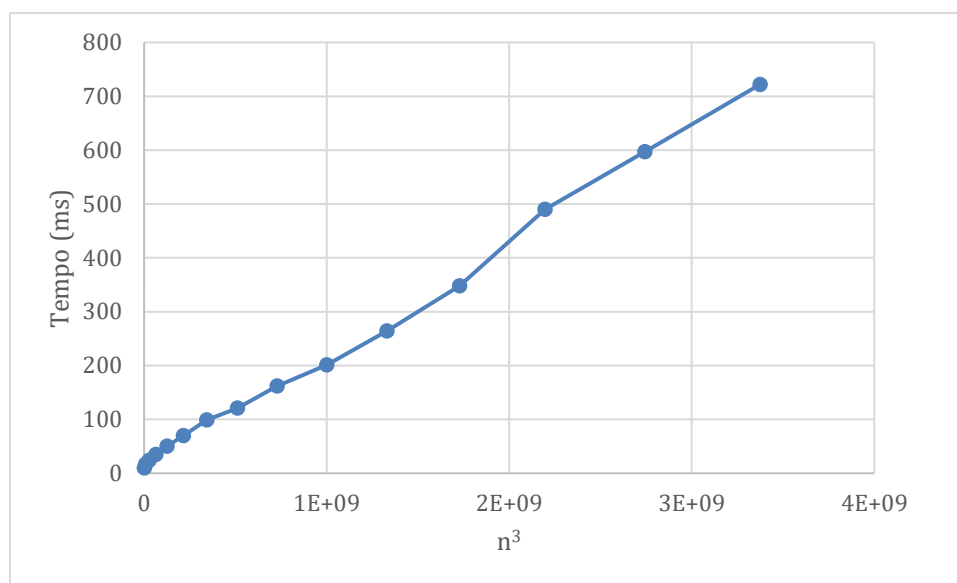
**Grupo:** al041

**Aluno(s):** Gonçalo Guerreiro (95581) e Vasco Correia (94188)

---

## Avaliação Experimental dos Resultados

Para verificar a complexidade estimada, foi testado o programa com vários grafos gerados utilizando o programa fornecido, com número de “processos” ( $n$ ) a variar entre 100 e 1500, com intervalos de 100. Ao gerar um gráfico do tempo de execução em função de  $n^3$ , verificou-se que o gráfico tem crescimento linear, observando-se assim que a razão entre os tempos de execução e  $n^3$  é aproximadamente constante, concluindo-se assim que a complexidade estimada está correta.



O programa foi testado num computador com as seguintes especificações:

- Processador de 8 cores com 2.00 Ghz
- 16 Gb de Memória RAM
- Sistema Operativo Linux(Ubuntu 20.04)