# Relatório 1º projecto ASA 2023/2024

**Grupo:** TP027
**Aluno(s):** Francisco Uva (106340) e Pedro Pais (107482)

---

## Descrição do Problema e da Solução

O código aborda um problema de otimização relacionado ao corte de peças retangulares, visando encontrar a melhor configuração para obter o maior valor total das peças selecionadas. Utilizando programação dinâmica, a solução preenche a matriz "piecesValues" com os valores máximos possíveis para cada combinação de dimensões das peças. A função "obtainMaxSellValue" calcula o valor máximo considerando cortes verticais e horizontais usando a matriz auxiliar "maxValues". A função "obtainMaxCutValue" determina os valores máximos ao considerar diferentes cortes em uma direção específica. O código percorre eficientemente todas as combinações possíveis de cortes, resultando no valor máximo total das peças. A estratégia adotada é sistemática, evitando redundâncias computacionais, garantindo eficiência na busca pela solução ideal. A utilização de matrizes auxiliares facilita o armazenamento de resultados intermediários, contribuindo para uma implementação clara e eficiente do algoritmo de otimização.

## Análise Teórica

Função recursiva da solução proposta.

Inserir aqui o pseudo código de muito alto nível a indicar a complexidade de cada etapa da solução proposta, e a complexidade total.

Exemplo:

- Leitura dos dados de entrada: simples leitura do input, com ciclos a depender linearmente de n (número de peças). Logo, O(n).
- Inicialização da matriz “piecesValues” com tamanho (X + 1) x (Y + 1), portanto, a complexidade é O(X * Y).
- Aplicação do algoritmo indicado para cálculo da função recursiva. Dentro da função “obtainMaxSellValue”, há dois loops aninhados que percorrem a matriz “maxValues” (X x Y). Para cada célula nessa matriz, a função “obtainMaxCutValue” é chamada, que possui dois loops independentes (um de X/2 e outro de Y/2). O loop interno de “obtainMaxCutValue” executa até X/2 ou Y/2, respetivamente. Considerando que a matriz tem dimensões X x Y, o pior caso seria metade dessas dimensões para cada loop interno. Portanto, a complexidade é O(X * Y * (X + Y)).
- Apresentação dos dados. A saída é uma única operação que imprime o resultado, logo O(1).

Complexidade global da solução:

A maior complexidade é a do passo 3, portanto, a complexidade global da solução proposta é aproximadamente O(X * Y * (X + Y)). Isso é derivado principalmente do processo de preenchimento das matrizes e da aplicação do algoritmo recursivo para calcular o valor máxim.

# Relatório 1º projecto ASA 2023/2024

**Grupo:** TP027

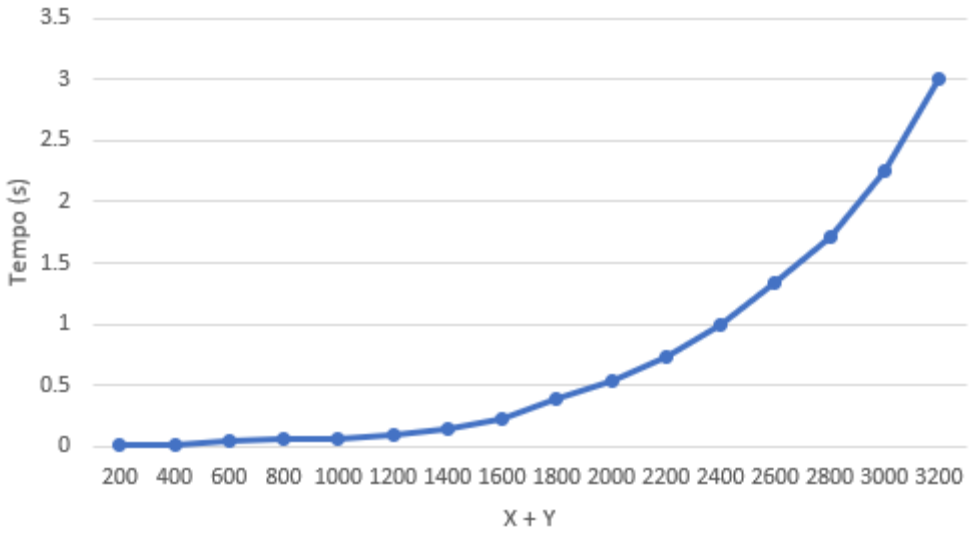
**Aluno(s):** Francisco Uva (106340) e Pedro Pais (107482)

---

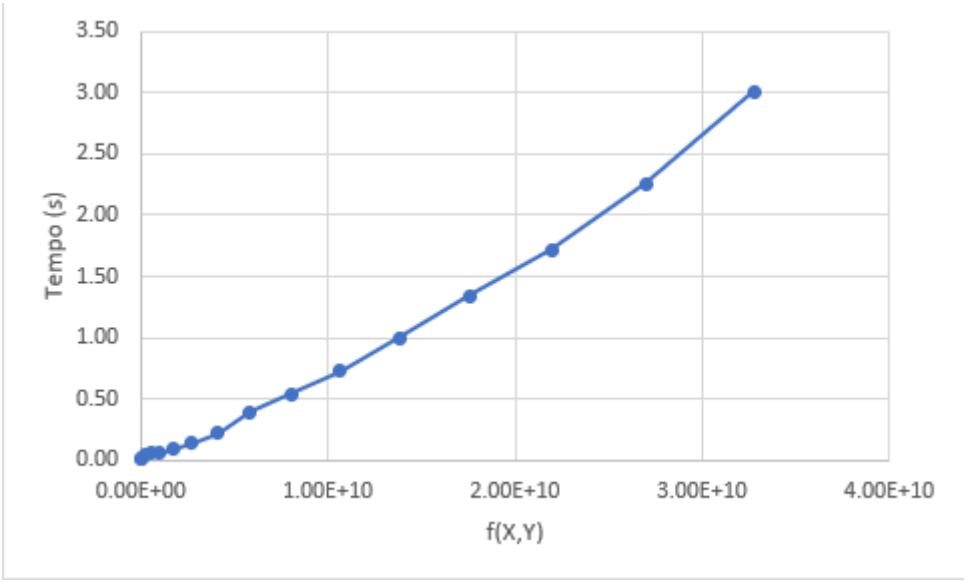
**Avaliação Experimental dos Resultados**

Foram geradas 16 instâncias, de tamanho incremental, e foi medido o tempo de execução para cada instância.

Gerou-se o gráfico do tempo (eixo do YYs) em função do tamanho das instâncias de entrada (eixo dos XXs).




O tempo de execução não é linear nas dimensões da chapa. Assim, vamos pôr o eixo dos XX a variar com a quantidade prevista pela análise teórica; exemplo: se a análise teórica for O(f(X, Y)), o tempo deve ser colocado em função de f(X, Y).




Ao mudarmos o eixo dos XX para f(X, Y), vemos que temos uma relação linear com os tempos no eixo dos YY, confirmando que a nossa implementação está de acordo com a análise teórica de O(f(X, Y)).