Relatório 1º Projeto ASA 2023/2024

Grupo: AL116

Aluno: Rodrigo Freire (106485)

Descrição do Problema e da Solução

Dado uma chapa de mármore retangular com dimensões $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle$ e uma lista com $\boldsymbol{N}$ tipos de peças $\boldsymbol{p_i}$ de tamanho $\boldsymbol{a_i} * \boldsymbol{b_i}$, $\boldsymbol{i} \in \{\boldsymbol{1}, .., \boldsymbol{N}\}$ com valor $\boldsymbol{v_i}$, pretende-se calcular o valor máximo $\boldsymbol{V}$ que pode ser obtido ao cortar a chapa horizontalmente ou verticalmente em peças correspondentes às especificadas pela lista. Cada peça pode ser produzida mais que uma vez, ou não produzida de todo se dessa escolha resultar um maior valor total. ($\boldsymbol{\beta_i}$ vezes, $\boldsymbol{\beta_i} \geq 0$)

$$V = \max \sum_{i=1}^{N} v_i \beta_i \ : \sum_{i=1}^{N} a_i b_i \beta_i \leq XY \ and \ \beta_i \geq 0 \ and \ a_i, b_i, X, Y \geq 1$$

A solução que eu proponho utiliza programação dinâmica e *memoization* de modo a otimizar o problema recursivo. Cada subproblema define-se por uma sub-chapa de dimensões $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ e a estrutura de *memoization* escolhida foi um vetor 1D que guarda o melhor valor para uma sub-chapa $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. O resultado de cada subproblema é dado pelo melhor valor obtido entre um corte horizontal ou vertical numa certa linha ou coluna (respetivamente).

Análise Teórica

A função recursiva $V(x, y)$ que permite calcular o resultado dá-se da seguinte forma:

$$let \ PieceCut(x, y, a_i, b_i) = \begin{cases} \max\big(V(x, b_i) + V(x, y - b_i), V(a_i, y) + V(x - a_i, y) \big) & if \ \ x \geq a_i \ \wedge y \geq b_i \\ 0 & otherwise \end{cases}$$

$$V(x, y) = \begin{cases} 0 & if \ x \leq 0 \vee y \leq 0 \\ \max\big(v_i, PieceCut(x, y, a_i, b_i), PieceCut(x, y, b_i, a_i)\big) & if \ x = a_i \wedge y = b_i \\ \max\big(PieceCut(x, y, a_i, b_i), PieceCut(x, y, b_i, a_i)\big) & otherwise \end{cases}$$

Pseudocódigo:

FUNCTION *computeCut*$(table, C, L)$*:*
  IF $C \leq 1$ **OR** $L \leq 1$:
    RETURN 0
  LET bestCut := 0
  FOR C_i from $1..\lfloor \frac{C}{2} \rfloor$:
    bestCut = max (bestCut, $table[L][C_i] + table[L][C - C_i]$)
  FOR L_i from $1..\lfloor \frac{L}{2} \rfloor$:
    bestCut = max (bestCut, $table[L_i][C] + table[L - L_i][C]$)
  RETURN bestCut

FUNCTION *solveBestValue*$(table, C, L)$*:*
  FOR each line L in table:
    FOR each column C in table:
      LET bestValue = max ($table[C][L], computeCut(table, C, L)$
      IF rotated subsheet exists:
        $table[C][L]$ = bestValue

  RETURN last value of table

```
FUNCTION main():
  READ X, Y, N     // Dimensão X, Dimensão Y, Número de Peças
  LET result := 0
  LET C := max(X,Y), L := min(X,Y)

  LET table[0..L][0..C] be a new Vector filled with 0.
  IF N > 0:
    FOR each piece P_i, i ∈ {1..N}:
      READ a_i, b_i, v_i
      table[b_i][a_i] = max (table[b_i][a_i], v_i)
      IF rotated piece P_i fits:
        table[a_i][b_i] = max (table[a_i][b_i], v_i)

    result = solveBestValue(table, C, L)
  RETURN result
```

Complexidades:

- Leitura dos dados de entrada e inserção do preço da peça lido na tabela de *memoization*. $\Theta(N)$
- Iteração sobre as colunas da tabela de *memoization*. $O(C)$
- Iteração sobre as linhas da tabela de *memoization*. $O(L)$
- Cálculo do melhor valor para sub-chapa $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. $O\left(\frac{C}{2} + \frac{L}{2}\right) = O(C + L)$
- Apresentação de resultados. $\Theta(1)$

Complexidade global da solução:

$$
\begin{aligned}
T(C, L, N) &= \Theta(N) + O(C) * O(L) * O(C + L) + \Theta(1) \\
&= \Theta(N) + \Theta(1) + O(C^2 L + L^2 C) \\
&= O(C^3 + C^3) \qquad (No\ pior\ caso\ C = L\ ) \\
&= O(C^3)
\end{aligned}
$$

Relatório 1º Projeto ASA 2023/2024

Grupo: AL116

Aluno(s): Rodrigo Freire (106485)

Avaliação Experimental dos Resultados

Como referi antes, o pior caso dá-se quando a chapa é quadrangular de forma que

$$max(X,Y) = min(X,Y) \Leftrightarrow \mathrm{C} = \mathrm{L}$$

Para comprovar a minha análise teórica decidi gerar os seguintes gráficos.
(Fig.1) Tempo (sec.) em função dos argumentos de entrada $X + Y = C + L = 2C$
(Fig.2) Tempo (sec.) em função da complexidade da solução $O(C^3)$

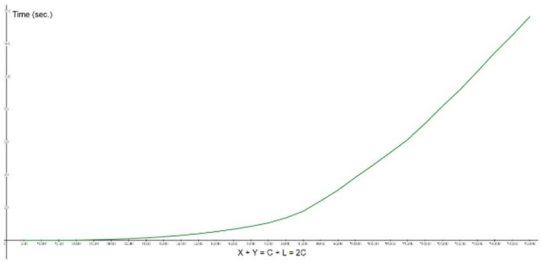

Fig.1

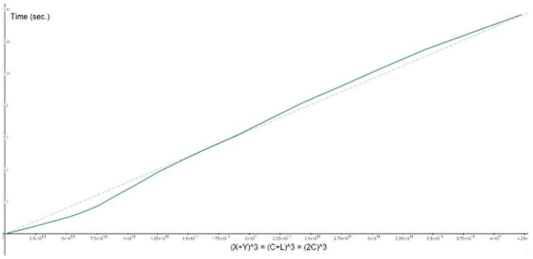

Fig.2

Foram geradas 30 instâncias de chapas quadrangulares de tamanho inicial $(X+Y)$ = 500, com um incremento de 500 por cada instância. A instância máxima tem tamanho $(X+Y)$ = 15000

Observamos que o gráfico (Fig.1) não produz tempo linear. Pelo gráfico (Fig.2) que se encontra em função da complexidade da solução reparamos que este produz uma relação linear.

Concluímos então que a implementação da solução se encontra em conformidade com a análise teórica.