Recursive Tiling

Contents

1 Basics

Aim of this document is to describe recursive algorithms to construct arbitrary big matrices $A^\star \in \{0, \dots, k^\star\}^{n^\star \times m^\star}$ from one (or more) significantly smaller matrices $A_0, \dots, A_\ell$ with $A_i \in \{0, \dots, k_i\}^{n_i \times m_i}$.

In general, we will have $k_i = k_j \leqslant k^\star$, $n_i = n_j$ and $m_i = m_j$ $\forall i, j$.

After constructing $A^\star$, we use a mapping from $\{0, \dots, k^\star\}$ to a set of colors and save the matrix as an image.

2 Notation

- As noted above, $A_0, \dots, A_i, \dots, A_\ell$ are the base matrices that are the input to our algorithms. n_i and m_i are the corresponding dimensions of matrix A_i.
- In constrast, $A^{(i)}$ will describe the matrix we have at the i-th iteration of the algorithm. Consequently, $n^{(i)}$ and $m^{(i)}$ are the dimensions of $A^{(i)}$.
- In order to not confuse the indices of a matrix from a set of matrices (e. g. A_i from $A_0, \dots, A_k$) with the indices of the elements of A_i, we will sometimes use the programming-language-inspired notation of $A_i[x, y]$ to identify the element at position x, y of matrix A_i.

3 Recursive Tiling

This approach is inspired by [1] (see Chapter 6, Figure 6.25, it's called *block inflation* there). The basic idea is as follows: we have the slightly stronger requirements $n := n_i = n_j, m := m_i = m_j \forall i, j$ and $\ell = k_i = k^\star$ $\forall i$, i. e. all base matrices A_i have the same size and there are exactly as many base matrices as there are possible different matrix elements.

3.1 A recursive tiling step using inflation tiling

We start with an arbitrary matrix $A^{(0)} \in \{0, \dots, k^\star\}^{n_0 \times m_0}$ (e. g. we can use $A^{(0)} = A_0$). Then, in iteration i, $A^{(i+1)}$ is constructed from $A^{(i)}$ by repacing each entry $a_{x,y}$ of $A^{(i)}$ by the base matrix $A_{a_{x,y}}$. Thus, $A^{(i+1)}$ is a block matrix of $n^{(i)} \times m^{(i)}$ many blocks $B_{x,y}$ of size $n \times m$ each that is conform to the following simple rule:

$$B_{x,y} = A_j \Leftrightarrow A^{(i)}_{x,y} = j.$$

Example 1.
Let

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

$$A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and $A^{(0)} = A_0$. *Then after one iteration, we have*

$$A^{(1)} = \left(\begin{array}{c|c} A_1 & A_0 \\ \hline A_0 & A_1 \end{array}\right) = \left(\begin{array}{cc|cc} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right).$$

After the second iteration, we have

$$A^{(2)} = \left(\begin{array}{c|c|c|c} A_0 & A_1 & A_1 & A_0 \\ \hline A_1 & A_0 & A_0 & A_1 \\ \hline A_1 & A_0 & A_0 & A_1 \\ \hline A_0 & A_1 & A_1 & A_0 \end{array}\right)$$

$$= \left(\begin{array}{cc|cc|cc|cc} 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \hline 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right)$$

4 Outer Sum

This time, the requirements are a little less restricted. We can use a set of arbitrary many matrices $A_1, \ldots, A_\ell$ that can all have arbitrary sizes and elements, i. e. $A_i \in \{0, \ldots, k_i\}^{n_i \times m_i}$ (However, $k_i > k^\star$ does not make much sense for the algorithm).

4.1 Preliminaries

In the spirit of the Kronecker Product, we define an "*outer sum*", of matrices as follows: Let $A_1 \in \mathbb{R}^{n_1 \times m_1}, A_2 \in \mathbb{R}^{n_2 \times m_2}$ be two matrices. The *outer sum* of A_1 and A_2 is a block matrix $B = A_1 \boxplus A_2 \in \mathbb{R}^{n_1 \cdot n_2 \times m_1 \cdot m_2}$ that consits of $n_2 \times m_2$ many blocks of size $n_1 \times m_1$.

The elements of the block B_{i_x,i_y} are

$$B_{i_x,i_y}[j_x, j_y] = A_1[j_x, j_y] + A_2[i_x, i_y].$$

That is, the full matrix looks as follows:

$$B = A_1 \boxplus A_2 =$$
$$\left(\begin{array}{ccc|c|ccc} A_1[1,1] + A_2[1,1] & \ldots & A_1[1,m_1] + A_2[1,1] & & A_1[1,1] + A_2[1,m_2] & \ldots & A_1[1,m_1] + A_2[1,m_2] \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ A_1[n_1,1] + A_2[1,1] & \ldots & A_1[n_1,m_1] + A_2[1,1] & & A_1[n_1,1] + A_2[1,m_2] & \ldots & A_1[n_1,m_1] + A_2[1,m_2] \\ \hline & \vdots & & \ddots & & \vdots & \\ \hline A_1[1,1] + A_2[n_2,1] & \ldots & A_1[1,m_1] + A_2[n_2,1] & & A_1[1,1] + A_2[n_2,m_2] & \ldots & A_1[1,m_1] + A_2[n_2,m_2] \\ \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ A_1[n_1,1] + A_2[n_2,1] & \ldots & A_1[n_1,m_1] + A_2[n_2,1] & & A_1[n_1,1] + A_2[n_2,m_2] & \ldots & A_1[n_1,m_1] + A_2[n_2,m_2] \end{array}\right)$$

4.2 A recursive tiling step using the outer sum

This time, we have to start with $A^{(0)} = A_0$ (alternatively, we could start with any arbitrary matrix of arbitrary size, this would add a global offset to the result).

In iteration i we compute $A^{(i)}$ as the outer sum of $A^{(i-1)}$ and $A_{i \bmod \ell}$. Note that due to the summing of elements, the absolute values of the matrix elements of B might exceed our maximum allowed absolute value $k^\star$. Therefore we apply the modulo-operation mod $(k^\star + 1)$ on each element of $A^{(i)}$ afterwards. Thus, the full iteration step is

$$A^{(i)} = (A^{(i-1)} \boxplus A_{i \bmod \ell}) \bmod (k^\star + 1).$$

Note that is sufficient to apply the outer modulo only once after the last iteration.

Example 2.
We will use the same matrices from Example 1, that is

$$A_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$
$$A_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and $A^{(0)} = A_0$. *With* $k^\star = 2$, *we have*

$$\begin{aligned}
A^{(1)} &= (A_0 \boxplus A_1)\ mod\ k^\star \\
&= \left(\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \boxplus \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \right)\ mod\ 2 \\
&= \left(\begin{array}{c|c} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 0 & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 1 \\ \hline \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 1 & \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + 0 \end{array} \right)\ mod\ 2 \\
&= \left(\begin{array}{cc|cc} 1 & 0 & 2 & 1 \\ 0 & 1 & 1 & 2 \\ \hline 2 & 1 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{array} \right)\ mod\ 2 \\
&= \left(\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ \hline 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array} \right)
\end{aligned}$$

Note that we are free to choose a larger $k^\star$, *this would yield a different result. The effect of* $k^\star$ *is increasing with each iteration.*

References

[1] M Baake and U Grimm. Aperiodic order, vol. 1: A mathematical invitation, cambridge u, 2013.