Technische Universität München
Lehrstuhl für Informatik V
Univ.-Prof. Dr. M. Bader
M.Sc. J.-M. Gallard, M.Sc. M. Obersteiner

SS 2018
Worksheet 13
04.07.2018

Algorithms of Scientific Computing

Arithmetization of Space-Filling Curves and Cache Efficiency

Exercise 1: Arithmetization of the Hilbert Curve

a) Initially we calculate the decimal places of the numbers and get:

$$\begin{aligned} \frac{1}{8} &= 0_4.02 \\ \frac{1}{3} &= 0_4.11111111\ldots \end{aligned}$$

So, for $h\left(\frac{1}{8}\right)$ we get:

$$\begin{aligned} h\begin{pmatrix}\frac{1}{8}\end{pmatrix} &= H_0 \circ H_2 \begin{pmatrix}0\\0\end{pmatrix} = H_0\left(\begin{pmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{pmatrix}\begin{pmatrix}0\\0\end{pmatrix} + \begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix}\right) \\ &= H_0\begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix} = \begin{pmatrix}0 & \frac{1}{2}\\ \frac{1}{2} & 0\end{pmatrix}\begin{pmatrix}\frac{1}{2}\\ \frac{1}{2}\end{pmatrix} = \begin{pmatrix}\frac{1}{4}\\ \frac{1}{4}\end{pmatrix} \end{aligned}$$

The calculation of $h\left(\frac{1}{3}\right)$ turns out to be much more complicated, since we need to find the following limit:

$$h\begin{pmatrix}\frac{1}{3}\end{pmatrix} = h(0_4.1111\ldots) = H_1 \circ H_1 \circ \ldots \begin{pmatrix}0\\0\end{pmatrix} = \lim_{n\to\infty} H_1^n \begin{pmatrix}0\\0\end{pmatrix}$$

So, we will write the operator H_1 in matrix-vector form:

$$H_1 = \underbrace{\begin{pmatrix}\frac{1}{2} & 0\\ 0 & \frac{1}{2}\end{pmatrix}}_{=:A_1}\underbrace{\begin{pmatrix}x\\y\end{pmatrix}}_{=:v} + \underbrace{\begin{pmatrix}0\\ \frac{1}{2}\end{pmatrix}}_{=:b_1} = A_1 v + b_1.$$

From this, we get

$$\begin{aligned} H_1^2 v &= A_1(A_1 v + b_1) + b_1 = A_1^2 v + A_1 b_1 + b_1 \\ H_1^3 v &= A_1(A_1^2 v + A_1 b_1 + b_1) + b_1 = A_1^3 v + A_1^2 b + A_1 b_1 + b_1 \\ \vdots &= \quad \vdots \\ H_1^n v &= A_1^n v + A_1^{n-1} b_1 + \ldots + A_1 b_1 + b_1 \end{aligned}$$

For the term $A_1^{n-1}b_1 + \ldots + A_1 b_1 + b_1$ we use a trick, similar to that used for geometric progressions:

$$\begin{aligned}(I - A_1)\left(A_1^{n-1}b_1 + \ldots + A_1 b_1 + b_1\right) &= A_1^{n-1}b_1 + \ldots + A_1 b_1 + b_1 \\ &\quad -A_1^n b_1 - A_1^{n-1}b_1 - \ldots - A_1 b_1 \\ &= b_1 - A_1^n b_1 = (I - A_1^n)\, b_1\end{aligned}$$

This leads to

$$A_1^{n-1}b_1 + \ldots + A_1 b_1 + b_1 = (I - A_1)^{-1}(I - A_1^n)\, b_1.$$

which renders to

$$\begin{aligned}\lim_{n\to\infty} H_1^n v &= \lim_{n\to\infty}(A_1^n v + \underbrace{A_1^{n-1}b_1 + \ldots + A_1 b_1 + b_1}_{(I-A_1)^{-1}(I-A_1^n)b_1}) \\ &= \lim_{n\to\infty}(\underbrace{A_1^n}_{\to 0} v + (I - A_1)^{-1}(I - \underbrace{A_1^n}_{\to 0})b_1) \\ &= (I - A_1)^{-1} b_1\end{aligned}$$

I.e. we get the value $h\left(\frac{1}{3}\right) = \lim_{n\to\infty} H_1^n v = (I - A_1)^{-1} b_1$. We can compute the inverse of $(I - A_1)$ easily because it is a diagonal matrix and obtain the final result

$$\begin{aligned}h\left(\frac{1}{3}\right) &= \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \frac{1}{2} \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 1 \end{pmatrix}.\end{aligned}$$

b)

$$\begin{aligned}\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} &= h\left(\frac{1}{2}\right) = h\left(\frac{1}{6}\right) = h\left(\frac{5}{6}\right) \\ \begin{pmatrix} 0 \\ 0 \end{pmatrix} &= h(0)\end{aligned}$$

Exercise 2: Arithmetization of the Peano Curve

Analogously to the arithmetization of the Hilbert curve, we assume that the parameter t is given on the basis 9. $t = 0_9.n_1 n_2 n_3 n_4 \ldots$. Now we are looking for the operators $P_0, \ldots, P_8$, so that

$$p(0_9.n_1 n_2 n_3 n_4 \ldots) = P_{n_1} \circ P_{n_2} \circ P_{n_3} \circ P_{n_4} \circ \cdots \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

The operators are given as

$$P_0 \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{3}x \\ \frac{1}{3}y \end{pmatrix}$$

$$
\begin{aligned}
P_1\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} -\frac{1}{3}x + \frac{1}{3} \\ \frac{1}{3}y + \frac{1}{3} \end{pmatrix} \\
P_2\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{3}x \\ \frac{1}{3}y + \frac{2}{3} \end{pmatrix} \\
P_3\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{3}x + \frac{1}{3} \\ -\frac{1}{3}y + 1 \end{pmatrix} \\
P_4\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} -\frac{1}{3}x + \frac{2}{3} \\ -\frac{1}{3}y + \frac{2}{3} \end{pmatrix} \\
P_5\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{3}x + \frac{1}{3} \\ -\frac{1}{3}y + \frac{1}{3} \end{pmatrix} \\
P_6\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{3}x + \frac{2}{3} \\ \frac{1}{3}y \end{pmatrix}. \\
P_7\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} -\frac{1}{3}x + 1 \\ \frac{1}{3}y + \frac{1}{3} \end{pmatrix} \\
P_8\begin{pmatrix} x \\ y \end{pmatrix} &= \begin{pmatrix} \frac{1}{3}x + \frac{2}{3} \\ \frac{1}{3}y + \frac{2}{3} \end{pmatrix}
\end{aligned}
$$

For the Peano functions see the attached Python program.

Exercise 3: Cache Efficiency

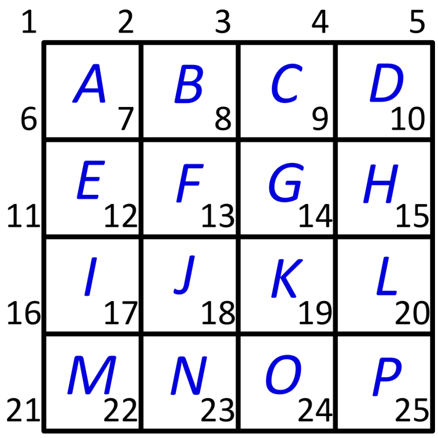

Figure 1: Labelling of the grids used in this solution

Table columns: Cell currently read, cache status before the read, element required, cache misses

Default traversal

Traversal order: A B C D E F G H I J K L M N O P

A	-	-	-	-	-	-	-	1 2 6 7	4
B	1	2	6	7	-	-	-	2 3 7 8	2
C	1*	2	6*	7	3	8	-	3 4 8 9	2
D	9	2*	6**	7*	3	8	4	4 5 9 10	2
E	9	10	5	7**	3*	8*	4	6 7 11 12	4
F	9*	10*	5*	6	7	11	12	7 8 12 13	2
G	13	10**	8	6	7	11*	12	8 9 13 14	2
H	13	9	8	6*	7*	14	12*	9 10 14 15	2
I	13*	9	8*	10	15	14	12**	11 12 16 17	4
J	16	17	12	10*	15*	14*	11	12 13 17 18	2
K	16*	17	12	13	15*	18	11*	13 14 18 19	2
L	16**	17*	12*	13	19	18	14	14 15 19 20	2
M	15	17**	20	13*	19	18*	14	16 17 21 22	4
N	15*	16	20*	17	19*	21	22	17 18 22 23	2
O	18	16*	20**	17	23	21*	22	18 19 23 24	2
P	18	24	19	17*	23	21**	22*	19 20 24 25	2
-	18*	24	19	25	23*	20	22**	-	-

Total cache misses: 40

Switch-Back traversal

Traversal order: A B C D H G F E I J K L P O N M

A	-	-	-	-	-	-	-	1 2 6 7	4
B	1	2	6	7	-	-	-	2 3 7 8	2
C	1*	2	6*	7	3	8	-	3 4 8 9	2
D	9	2*	6**	7*	3	8	4	4 5 9 10	2
H	9	10	5	7**	3*	8*	4	9 10 14 15	2
G	9	10	5*	14	15	8**	4*	8 9 13 14	1
F	9	10*	5**	14	15*	8	13	7 8 12 13	2
E	9*	12	7	14*	15**	8	13	6 7 11 12	2
I	11	12	7	14**	6	8*	13*	11 12 16 17	2
J	11	12	7*	16	6*	17	13**	12 13 17 18	1
K	11*	12	7**	16*	18	17	13	13 14 18 19	2
L	19	12*	14	16**	18	17*	13	14 15 19 20	2
P	19	20	14	15	18*	17**	13*	19 20 24 25	2
O	19	20	14*	15*	18**	24	25	18 19 23 24	1
N	19	20*	23	15**	18	24	25*	17 18 22 23	2
M	19*	22	23	17	18	24*	25**	16 17 21 22	2
-	21	22	23*	17	18*	24**	16	-	-

Total cache misses: 31

Hilbert traversal

Traversal order: A B F E I M N J K O P L H G C D

A	-	-	-	-	-	-	-	1 2 6 7	4
B	1	2	6	7	-	-	-	2 3 7 8	2
F	1*	2	6*	7	3	8	-	7 8 12 13	2
E	13	2*	6**	7	3*	8	12	6 7 11 12	1
I	13*	11	6	7	3**	8*	12	11 12 16 17	2
M	13**	11	6*	7*	16	17	12	16 17 21 22	2
N	21	11*	22	7**	16	17	12*	17 18 22 23	2
J	21*	23	22	18	16*	17	12**	12 13 17 18	1
K	21**	23*	22*	18	13	17	12	13 14 18 19	2
O	14	23**	19	18	13	17*	12*	18 19 23 24	1
P	14*	23	19	18	13*	17**	24	19 20 24 25	2
L	14**	23*	19	18*	25	20	24	14 15 19 20	1
H	14	23**	19	15	25*	20	24*	9 10 14 15	2
G	14	9	19*	15	25**	20*	10	8 9 13 14	2
C	14	9	13	15*	8	20**	10*	3 4 8 9	2
D	14*	9	13*	15**	8	3	4	4 5 9 10	2
-	14**	9*	10	5	8	3*	4	-	-

Total cache misses: 30