The SMAWK Algorithm

Lawrence L. Larmore UNLV

1 Introduction

The SMAWK algorithm finds all row minima of a totally monotone matrix, such as the one below. The name “SMAWK” is an acronym consisting of the first letters of the last names of the five developers of the algorithm. [1]

Monotonicity. A 2×2 matrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is defined to be *monotone* if the following two conditions hold:

1. If $c < d$, then $a < b$.
2. If $c = d$, then $a \leq b$.

A rectangular matrix is defined to be *totally monotone* if every 2×2 submatrix is monotone. Recall that a submatrix does not have to consist of adjacent rows or columns. For example, $\begin{bmatrix} 28 & 37 \\ 44 & 33 \end{bmatrix}$ is a submatrix of the 9×18 matrix shown below, obtained by taking the 3rd and 6th rows and the 7th and 9th columns.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	25	21	13	10	20	13	19	35	37	41	58	66	82	99	124	133	156	178
2	42	35	26	20	29	21	25	37	36	39	56	64	76	91	116	125	146	164
3	57	48	35	28	33	24	28	40	37	37	54	61	72	83	107	113	131	146
4	78	65	51	42	44	35	38	48	42	42	55	61	70	80	100	106	120	135
5	90	76	58	48	49	39	42	48	39	35	47	51	56	63	80	86	97	110
6	103	85	67	56	55	44	44	49	39	33	41	44	49	56	71	75	84	96
7	123	105	86	75	73	59	57	62	51	44	50	52	55	59	72	74	80	92
8	142	123	100	86	82	65	61	62	50	43	47	45	46	46	58	59	65	73
9	151	130	104	88	80	59	52	49	37	29	29	24	23	20	28	25	31	39

The algorithm uses two reductions, which are applied alternately and recursively. The first reduction, which we call INTERPOLATE, reduces the problem of finding the row minima of an $n \times m$ totally monotone matrix to the problem of finding the row minima of a $\left\lfloor \frac{n}{2} \right\rfloor \times m$ totally monotone matrix. We use this reduction when $m \leq n$.

The second reduction, which we call REDUCE, reduces the problem of finding the row minima of an $n \times m$ totally monotone matrix to the problem of finding the row minima of an $n \times m'$ totally monotone matrix for some $m' \leq n$. We use this reduction when $m > n$.

2 REDUCE

Since our example matrix has more columns than rows, we will start with REDUCE.

Consider an $n \times m$ totally monotone matrix M. Write $M[i,j]$ for the entry of M in the ith row and the j^{th} column. We write C_j for the jth column of M.

The fundamental idea of REDUCE is to compare two entries in the same row, and then to use the monotonicity property to eliminate part of one of the two columns. More specifically, if $M[i,j] < M[i,k]$, for $j < k$, we can eliminate all entries of the form $M[\ell,j]$, for $\ell \geq i$, from consideration. On the other had, if $M[i,j] > M[i,k]$, we can eliminate all entries of the form $M[\ell,k]$, for $\ell \leq i$. If they are equal, we get our choice.

Our goal is to delete at least $m - n$ columns of M. We then continue SMAWK on the matrix consisting of the surviving columns.

We maintain a stack S of *surviving columns*. Each column on the stack has a *head* which is its topmost *surviving entry*. All entries of a column on the stack which are above the head are deleted. Initially, S is empty.

We process the columns one at a time, from left to right. When we process the column C_j, we execute the following steps.

1. If S is empty, push the C_j onto the stack, and mark its first entry *i.e.,* the entry in row 1, as its head.
2. Otherwise, iterate the following loop until either S is empty or C_j is *defeated*.
   (a) Let C_ℓ be the top entry of S. Let $M[i,\ell]$ be the head of C_ℓ.
   (b) Compare $M[i,\ell]$ and $M[i,j]$. We will then execute one of the following three steps.
      i. If $M[i,j] \leq M[i,\ell]$, then pop C_ℓ off S. C_ℓ has now been eliminated. (The loop continues, as C_j attacks the next column on the stack.)
      ii. If $M[i,j] > M[i,\ell]$ and $i = n$, then C_j is defeated and eliminated. Exit the loop.
      iii. If $M[i,j] > M[i,\ell]$ and $i < n$, then C_j is defeated, but still survives. Push C_j onto S, and mark $M[i+1,j]$ as its head. Exit the loop.

When all columns have been processed, the heads of all surviving columns are in different rows. Thus, there can be at most n surviving columns. The reduced matrix consists of all surviving columns and all rows. The row minima of the reduced matrix are the row minima of M.

2.1 An Example Execution of REDUCE

We will walk through REDUCE for the matrix given in the introduction.

We first push C_1 onto S. S consists only of C_1, and the head of that columns is enclosed in a square.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	$\boxed{25}$	21	13	10	20	13	19	35	37	41	58	66	82	99	124	133	156	178
2	42	35	26	20	29	21	25	37	36	39	56	64	76	91	116	125	146	164
3	57	48	35	28	33	24	28	40	37	37	54	61	72	83	107	113	131	146
4	78	65	51	42	44	35	38	48	42	42	55	61	70	80	100	106	120	135
5	90	76	58	48	49	39	42	48	39	35	47	51	56	63	80	86	97	110
6	103	85	67	56	55	44	44	49	39	33	41	44	49	56	71	75	84	96
7	123	105	86	75	73	59	57	62	51	44	50	52	55	59	72	74	80	92
8	142	123	100	86	82	65	61	62	50	43	47	45	46	46	58	59	65	73
9	151	130	104	88	80	59	52	49	37	29	29	24	23	20	28	25	31	39

In the next step, C_2 challenges and eliminates C_1. Since S is now empty, C_2 is pushed onto the stack, with head $M[1,2]$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		$\boxed{21}$	13	10	20	13	19	35	37	41	58	66	82	99	124	133	156	178
2		35	26	20	29	21	25	37	36	39	56	64	76	91	116	125	146	164
3		48	35	28	33	24	28	40	37	37	54	61	72	83	107	113	131	146
4		65	51	42	44	35	38	48	42	42	55	61	70	80	100	106	120	135
5		76	58	48	49	39	42	48	39	35	47	51	56	63	80	86	97	110
6		85	67	56	55	44	44	49	39	33	41	44	49	56	71	75	84	96
7		105	86	75	73	59	57	62	51	44	50	52	55	59	72	74	80	92
8		123	100	86	82	65	61	62	50	43	47	45	46	46	58	59	65	73
9		130	104	88	80	59	52	49	37	29	29	24	23	20	28	25	31	39

In the next two steps, C_3 challenges and eliminates C_2, and then C_4 challenges and eliminates C_3. The resulting matrix:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$	20	13	19	35	37	41	58	66	82	99	124	133	156	178
2				20	29	21	25	37	36	39	56	64	76	91	116	125	146	164
3				28	33	24	28	40	37	37	54	61	72	83	107	113	131	146
4				42	44	35	38	48	42	42	55	61	70	80	100	106	120	135
5				48	49	39	42	48	39	35	47	51	56	63	80	86	97	110
6				56	55	44	44	49	39	33	41	44	49	56	71	75	84	96
7				75	73	59	57	62	51	44	50	52	55	59	72	74	80	92
8				86	82	65	61	62	50	43	47	45	46	46	58	59	65	73
9				88	80	59	52	49	37	29	29	24	23	20	28	25	31	39

In the next step, C_5 challenges C_4 and is defeated. It is then pushed onto S, with head $M[2,5]$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$		13	19	35	37	41	58	66	82	99	124	133	156	178
2				20	$\boxed{29}$	21	25	37	36	39	56	64	76	91	116	125	146	164
3				28	33	24	28	40	37	37	54	61	72	83	107	113	131	146
4				42	44	35	38	48	42	42	55	61	70	80	100	106	120	135
5				48	49	39	42	48	39	35	47	51	56	63	80	86	97	110
6				56	55	44	44	49	39	33	41	44	49	56	71	75	84	96
7				75	73	59	57	62	51	44	50	52	55	59	72	74	80	92
8				86	82	65	61	62	50	43	47	45	46	46	58	59	65	73
9				88	80	59	52	49	37	29	29	24	23	20	28	25	31	39

In the next step, C_6 challenges and eliminates C_5, since $21 \leq 29$. It then challenges C_4 and is defeated, since $13 > 10$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$			19	35	37	41	58	66	82	99	124	133	156	178
2				20		$\boxed{21}$	25	37	36	39	56	64	76	91	116	125	146	164
3				28		24	28	40	37	37	54	61	72	83	107	113	131	146
4				42		35	38	48	42	42	55	61	70	80	100	106	120	135
5				48		39	42	48	39	35	47	51	56	63	80	86	97	110
6				56		44	44	49	39	33	41	44	49	56	71	75	84	96
7				75		59	57	62	51	44	50	52	55	59	72	74	80	92
8				86		65	61	62	50	43	47	45	46	46	58	59	65	73
9				88		59	52	49	37	29	29	24	23	20	28	25	31	39

In the next step, C_7 challenges C_6 and is defeated, since $21 < 25$. In the following step, C_8 challenges C_7 and is defeated, since $28 < 40$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$					37	41	58	66	82	99	124	133	156	178
2				20		$\boxed{21}$			36	39	56	64	76	91	116	125	146	164
3				28		24	$\boxed{28}$		37	37	54	61	72	83	107	113	131	146
4				42		35	38	$\boxed{48}$	42	42	55	61	70	80	100	106	120	135
5				48		39	42	48	39	35	47	51	56	63	80	86	97	110
6				56		44	44	49	39	33	41	44	49	56	71	75	84	96
7				75		59	57	62	51	44	50	52	55	59	72	74	80	92
8				86		65	61	62	50	43	47	45	46	46	58	59	65	73
9				88		59	52	49	37	29	29	24	23	20	28	25	31	39

In the next step, C_9 challenges and eliminates C_8, and then challenges C_7 and is defeated. In the following step, C_{10} challenges and eliminates C_9, since $42 \leq 42$. Then, C_{10} challenges C_7 and is defeated.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$							58	66	82	99	124	133	156	178
2				20		$\boxed{21}$					56	64	76	91	116	125	146	164
3				28		24	$\boxed{28}$				54	61	72	83	107	113	131	146
4				42		35	38			$\boxed{42}$	55	61	70	80	100	106	120	135
5				48		39	42			35	47	51	56	63	80	86	97	110
6				56		44	44			33	41	44	49	56	71	75	84	96
7				75		59	57			44	50	52	55	59	72	74	80	92
8				86		65	61			43	47	45	46	46	58	59	65	73
9				88		59	52			29	29	24	23	20	28	25	31	39

In the next five steps, each new column is defeated in its first challenge, and gets pushed onto S, because $55 > 42$, $51 > 47$, $49 > 44$, $59 > 55$, and $58 > 46$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$												133	156	178
2				20		$\boxed{21}$										125	146	164
3				28		24	$\boxed{28}$									113	131	146
4				42		35	38			$\boxed{42}$						106	120	135
5				48		39	42			35	$\boxed{47}$					86	97	110
6				56		44	44			33	41	$\boxed{44}$				75	84	96
7				75		59	57			44	50	52	$\boxed{55}$			74	80	92
8				86		65	61			43	47	45	46	$\boxed{46}$		59	65	73
9				88		59	52			29	29	24	23	20	$\boxed{28}$	25	31	39

In the next step, C_{16} challenges and eliminates C_{15}, and then challenges C_{14} and is defeated.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$													156	178
2				20		$\boxed{21}$											146	164
3				28		24	$\boxed{28}$										131	146
4				42		35	38			$\boxed{42}$							120	135
5				48		39	42			35	$\boxed{47}$						97	110
6				56		44	44			33	41	$\boxed{44}$					84	96
7				75		59	57			44	50	52	$\boxed{55}$				80	92
8				86		65	61			43	47	45	46	$\boxed{46}$			65	73
9				88		59	52			29	29	24	23	20		$\boxed{25}$	31	39

In the last two steps, C_{17} and C_{18} each challenge C_{16} and are defeated. However, since the head of C_{16} is in the n^{th} row, these columns are eliminated and not pushed onto S.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				$\boxed{10}$														
2				20		$\boxed{21}$												
3				28		24	$\boxed{28}$											
4				42		35	38			$\boxed{42}$								
5				48		39	42			35	$\boxed{47}$							
6				56		44	44			33	41	$\boxed{44}$						
7				75		59	57			44	50	52	$\boxed{55}$					
8				86		65	61			43	47	45	46	$\boxed{46}$				
9				88		59	52			29	29	24	23	20		$\boxed{25}$		

The row minima of M must be located in the entries that have not been eliminated. They need not be the heads of the columns.

We continue the recursion by executing SMAWK on the square matrix consisting of the surviving columns:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				10		13	19			41	58	66	82	99		133		
2				20		21	25			39	56	64	76	91		125		
3				28		24	28			37	54	61	72	83		113		
4				42		35	38			42	55	61	70	80		106		
5				48		39	42			35	47	51	56	63		86		
6				56		44	44			33	41	44	49	56		75		
7				75		59	57			44	50	52	55	59		74		
8				86		65	61			43	47	45	46	46		59		
9				88		59	52			29	29	24	23	20		25		

Since the reduced matrix has at least as many rows as columns, the recursive application of SMAWK will begin with INTERPOLATE. Despite the fact that we know that almost half the reduced matrix could not contain any row minima, we do not use this information in the next step.

3 INTERPOLATE

We now explain the *interpolation* phase of SMAWK. We usually interpolate only if there are at least as many rows as columns.

As an example, we start with the output of the REDUCE example given in Section 2. We begin by eliminating every second row, starting with the first row:

				4		6	7			10	11	12	13	14		16		
2				20		21	25			39	56	64	76	91		125		
4				42		35	38			42	55	61	70	80		106		
6				56		44	44			33	41	44	49	56		75		
8				86		65	61			43	47	45	46	46		59		

We then execute SMAWK recursivly on this 4×9 matrix. When we find the row minima of this matrix, we will be able to compute the row minima of the 9×9 matrix by interpolation.

In order to show the actual recursive descent and ascent, we will now finish all the steps of SMAWK for our example.

4 Finishing Our Example

Apply REDUCE to the 4×9 matrix, in steps.

				4		6	7			10	11	12	13	14		16		
2				$\boxed{20}$		21	25			39	56	64	76	91		125		
4				42		35	38			42	55	61	70	80		106		
6				56		44	44			33	41	44	49	56		75		
8				86		65	61			43	47	45	46	46		59		

				4		6	7			10	11	12	13	14		16		
2				$\boxed{20}$			25			39	56	64	76	91		125		
4				42		$\boxed{35}$	38			42	55	61	70	80		106		
6				56		44	44			33	41	44	49	56		75		
8				86		65	61			43	47	45	46	46		59		

				4		6	7			10	11	12	13	14		16		
2				$\boxed{20}$						39	56	64	76	91		125		
4				42		$\boxed{35}$				42	55	61	70	80		106		
6				56		44	$\boxed{44}$			33	41	44	49	56		75		
8				86		65	61			43	47	45	46	46		59		

				4		6	7			10	11	12	13	14		16		
2				$\boxed{20}$							56	64	76	91		125		
4				42		$\boxed{35}$					55	61	70	80		106		
6				56		44				$\boxed{33}$	41	44	49	56		75		
8				86		65				43	47	45	46	46		59		

				4		6	7			10	11	12	13	14		16		
2				$\boxed{20}$								64	76	91		125		
4				42		$\boxed{35}$						61	70	80		106		
6				56		44				$\boxed{33}$		44	49	56		75		
8				86		65				43	$\boxed{47}$	45	46	46		59		

				4		6	7			10	11	12	13	14		16		
2				$\boxed{20}$									76	91		125		
4				42		$\boxed{35}$							70	80		106		
6				56		44				$\boxed{33}$			49	56		75		
8				86		65				43		$\boxed{45}$	46	46		59		

				4		6	7			10	11	12	13	14		16		
2				$\boxed{20}$														
4				42		$\boxed{35}$												
6				56		44				$\boxed{33}$								
8				86		65				43		$\boxed{45}$						

We now interpolate, starting the 4×4 result of the previous phase.

				4		6				10		12						
2				20		21				39		64						
4				42		35				42		61						
6				56		44				33		44						
8				86		65				43		45						

Eliminate the first and third rows.

				4		6				10		12						
4				42		35				42		61						
8				86		65				43		45						

We use REDUCE again, and obtain a 2×2 matrix.

						6				10								
4						$\boxed{35}$												
8						65				$\boxed{43}$								

At this point, we simply search each of the remaining rows for its minimum.

						6				10								
4						**35**												
8										**43**								

We now replace the previously eliminated rows, and indicate our search space to find the minimum of each of those rows.

				4		6				10		12						
2				20		21												
4						**35**												
6						44				33								
8										**43**								

We now do linear search to find the minima of those rows.

				4		6				10		12						
2				**20**														
4						**35**												
6										**33**								
8										**43**								

As we ascend out of the recursion, we replace the missing columns that we eliminated in the second REDUCE step, and the rows we eliminated in the first INTERPOLATE step. Those steps has actually been in suspension, and will now be completed.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	25	21	13	10	20	13	19	35	37	41	58	66	82	99	124	133	156	178
2				**20**														
3	57	48	35	28	33	24	28	40	37	37	54	61	72	83	107	113	131	146
4						**35**												
5	90	76	58	48	49	39	42	48	39	35	47	51	56	63	80	86	97	110
6										**33**								
7	123	105	86	75	73	59	57	62	51	44	50	52	55	59	72	74	80	92
8										**43**								
9	151	130	104	88	80	59	52	49	37	29	29	24	23	20	28	25	31	39

We now eliminate all but the intervals that must be searched.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	25	21	13	10														
2				**20**														
3				28	33	24												
4						**35**												
5						39	42	48	39	35								
6										**33**								
7										44								
8										**43**								
9										29	29	24	23	20	28	25	31	39

We now find the minima of all those intervals, using linear search.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1				**10**														
2				**20**														
3						**24**												
4						**35**												
5										**35**								
6										**33**								
7										**44**								
8										**43**								
9														**20**				

We have found all row minima. We show the original matrix with the row minima in bold face.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	25	21	13	**10**	20	13	19	35	37	41	58	66	82	99	124	133	156	178
2	42	35	26	**20**	29	21	25	37	36	39	56	64	76	91	116	125	146	164
3	57	48	35	28	33	**24**	28	40	37	37	54	61	72	83	107	113	131	146
4	78	65	51	42	44	**35**	38	48	42	42	55	61	70	80	100	106	120	135
5	90	76	58	48	49	39	42	48	39	**35**	47	51	56	63	80	86	97	110
6	103	85	67	56	55	44	44	49	39	**33**	41	44	49	56	71	75	84	96
7	123	105	86	75	73	59	57	62	51	**44**	50	52	55	59	72	74	80	92
8	142	123	100	86	82	65	61	62	50	**43**	47	45	46	46	58	59	65	73
9	151	130	104	88	80	59	52	49	37	29	29	24	23	**20**	28	25	31	39

References

[1] Alok Aggarwal, Maria M. Klawe, Shlomo Moran, Peter W. Shor, and Robert E. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2:195–208, 1987.