EECS587 HW4

author: Yilin Jia

date: 2024/11/6

Method

To find the second smallest number in 4 numbers. tmp0 is the mininum of maximum, which is possibly the second smallest or third smallest number. tmp1 is the maximum of mininum, which is possibly the second smallest or third smallest number. Therefore, the smaller one between them is exactly the second smallest number in 4 numbers.

```
tmp0 = min(max(A[i - 1][j - 1], A[i + 1][j + 1]), max(A[i + 1][j - 1], A[i - 1][j + 1]));
tmp1 = max(min(A[i - 1][j - 1], A[i + 1][j + 1]), min(A[i + 1][j - 1], A[i - 1][j + 1]));
tmp1 = min(tmp1, tmp0);
```

In the cuda version, instead of copy B to A in every iteration, I just roll the matrix(A->B->A->B...). This can void copying the data.

Result

(n,t)	type	Time(ms)	A[37][47]	Sum
(500,10)	serial	10.0	541.703	1.2147e+08
(500,10)	cuda	4.0	541.703	1.2147e+08
(1000,10)	serial	45.0	541.703	4.91725e+08
(1000,10)	cuda	4.6	541.703	4.91725e+08
(2000,10)	serial	192.0	541.703	1.97807e+09
(2000,10)	cuda	12.7	541.703	1.97807e+09

From the table, we can find that A[37][47] doesn't change when t is fixed. Beside, the sum increases when n goes up.
We can see the running time of serial version increased qudraticly while the cuda version doesn't change so much since the cuda can parallel compute the matrix.