① Parallel cyclic reduction (PCR) diagonally dominant matrices or symmetric and positive definite matrices

Tridiagonal Solvers

② Householder reduction → tridiagonal decomposition of any symmetric matrix

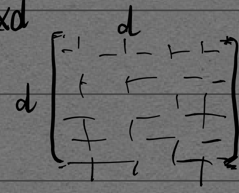
__global__ myk (   ) : <<< numBlocks, threadsPerBlock>>> → d的倍数 dxd

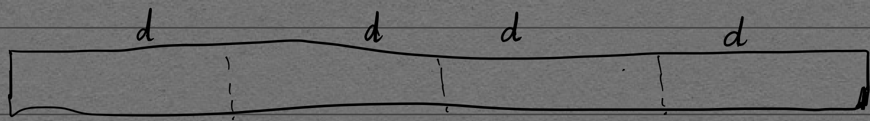
int tidx = threadIdx.x % d ; (0~d-1) d : dimension of matrix

int Qt = (threadIdx.x − tidx) / d ;

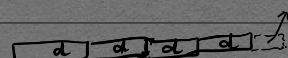
int gbx = Qt + blockIdx.x * (blockDim.x / d) ;

dxd





PCR_Device_Functions.cu



int tidx = threadIdx.x % d 求余 → tidx 是在每个block中

int Qt = (threadIdx.x − tidx) / d ; d + d + d + d 4个

int gbx = Qt + blockIdx.x * (blockDim.x / d) ;
↓ 每个block总的thread数.

这样得到的 gbx 全的的 index

$P = (d/2 + (d\%2)) * (tidx \% 2) + (int)\ tidx/2;$

$d = 8$, $P = \underbrace{(8/2 + (8\%2))} * (tidx \% 2) + (int)\ tidx/2;$

tidx % 2 ↓ threadIdx.x % d

example: blockDim.x = 16, d = 8

threadIdx	tidx	tidx % 2	P
0	0	0	0
1	1	1	4+0=4
2	2	0	1
3	3	1	4+1=5
4	4	0	2
5	5	1	4+2=6
6	6	0	3
7	7	1	4+3=7
8	0	0	0
9	1	1	4
10	2	0	1
11	3	1	5
12	4	0	2
13	5	1	6

13 - - - - 3

14 - - - 6 - - - - 0 - - - 3

15 - - - 7 - - - - 1 - - - 7

$P = (d/2 + (d\%2)) * (tidx\ \%\ 2) + (int)\ tidx/2;$

$d = 11,\quad P = (\underbrace{\overset{5}{11/2}}_{int} + \overset{1}{(11\%2)})\ \ * \ (tidx\ \%\ 2) + (int)\ tidx/2;$ with the bracketed sum $= 6$

$tidx \downarrow$ threadIdx.x % d

example: blockDim.x = 22, d = 11

threadIdx	tidx	tidx % 2	P
0	0	0	0
1	1	1	6 + 0 = 6
2	2	0	1
3	3	1	6 + 1 = 7
4	4	0	2
5	5	1	6 + 2 = 8
6	6	0	3
7	7	1	9
8	8	0	4
9	9	1	10
10	10	0	5
11	0	0	
12	1	1	
13	2	0	
14	3	1	
15	4	0	
16	5	1	
17	6	0	
18	7	1	
19	8	0	
20	9	1	
21	10	0	

$\overset{x_{i-2}}{\uparrow}\ \ \overset{x_{i-1}}{\uparrow}\ \ \overset{x_i}{\uparrow}$

$a_{i-1}\quad b_{i-1}\quad c_{i-1}$

$a_i \nearrow x_{i-1}\quad b_i \nearrow x_i\quad c_i \nearrow x_{i+1}$

$a_{i+1} \quad b_{i+1} \quad c_{i+1}$

$\downarrow \quad \downarrow \quad \downarrow$

$x_{i-1} \quad x_{i+1} \quad x_{i+2}$

$k_1 = \frac{a_i}{b_{i-1}}$ $\qquad a_i' = - \frac{a_{i-1} a_i}{b_{i-1}} = -a_{i-1} \times k_1$

i对应的行是不变的，但是对应的列有变化 — stride

$k_2 = \frac{c_i}{b_{i+1}}$ $\qquad c_i' = - \frac{c_{i+1} c_i}{b_{i+1}} = - c_{i+1} k_2$

$b_i' = b_i - c_{i-1} k_1 - a_{i+1} k_2$

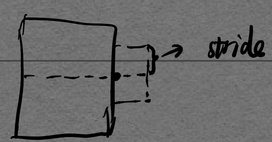
$d_i' = d_i - d_{i-1} k_1 - d_{i+1} k_2$

alist blist clist dlist xlist int_max int DMax

一个block，每一个 thread_id 对应一个 idx_row

row_max = DMax − 1

stride ※ → 控制



a1, b1, c1, d1,

k01, k21, c01, a21, d01, d21

next_or_ot = true; iter_max 最大迭代次数

next_stride = stride << 1; → 左进位符，二进制，相当于 2^n

if idx_row − stride < 0 → if it is the first line

{ a1 = 0, k01 = 0, c01 = 0, d01 = 0 }

// Determine if this line has died the bi-set

int pos = idx_row − 2* stride;

accum = 0

第一次：

stride = 1, $\Longrightarrow$ $\begin{bmatrix} d_1 & 0 & c_1' & & \\ 0 & d_2' & 0 & c_2' & \\ & & \ddots & & \ddots \end{bmatrix}$

stride = next_stride = 4 Dmax = 32

idx_row = 31, pos = 31 − 2 × 4 = 23

accum = 0;

for (size_t iter = 0; iter < 5; iter ++) {

① accum = 1
pos = 23 + 8

② accum = 2
pos =

③ accum = 3

$d_1 \; c_1$
$a_2 \; d_2 \; c_2$
$a_3 \; d_3 \; c_3$
$a_4 \; d_4 \; c_4$
$a_5 \; d_5 \; c_5$
$a_6 \; d_6 \; c_6$
$a_7 \; d_7 \; c_7$
$a_8 \; d_8 \; c_8$
$a_9 \; d_9 \; c_9$
$a_{10} \; d_{10} \; c_{10}$
$a_{11} \; d_{11}$

$d_1' \; 0 \; c_1'$
$0 \; d_2' \; 0 \; c_2'$
$a_3' \; 0 \; d_3' \; 0 \; c_3'$
$a_4' \; 0 \; d_4' \; 0 \; c_4'$
$a_5' \; 0 \; d_5' \; 0 \; c_5'$
$a_6' \; 0 \; d_6' \; 0 \; c_6'$
$a_7' \; 0 \; d_7' \; 0 \; c_7'$
$a_8' \; 0 \; d_8' \; 0 \; c_8'$
$a_9' \; 0 \; d_9' \; 0 \; c_9'$
$a_{10}' \; 0 \; d_{10}' \; 0$
$a_{11}' \; 0 \; d_{11}'$

1 - 7
2 - 8
3 - 9
4 - 10
5 - 11
6

1
3
5
7
9
11

2
4
6
8
10

$d_2 \; c_2$
$a_4 \; d_4 \; c_4$
$a_6 \; d_6 \; c_6$
$a_8 \; d_8 \; c_8$
$a_{10} \; d_{10}$

$\Downarrow$

$0 \; d_2'' \; 0 \; c_2''$
$0 \; d_4'' \; 0 \; c_4''$
$a_6'' \; 0 \; d_6'' \; 0 \; c_6''$
$a_8'' \; 0 \; d_8'' \; 0$

$a_8 \quad 0 \quad d_8 \quad 0$

$a_{10}'' \quad 0 \quad d_{10}''$

$\Downarrow$

$0 \quad d_2'' \quad 0 \quad c_2''$

$a_6'' \quad 0 \quad d_6'' \quad 0 \quad c_6''$

$0 \quad a_{10}'' \quad 0 \quad d_{10}''$

$0 \quad d_4'' \quad 0 \quad c_4''$

$a_8'' \quad 0 \quad d_8'' \quad 0$

$\Downarrow$

$0 \quad d_2'' \quad c_2''$

$a_6'' \quad d_6'' \quad c_6'' \quad 0$

$a_{10}'' \quad d_{10}'' \quad 0$

$d_4'' \quad a_4''$

$a_8'' \quad d_8''$

$\Downarrow$

$0 \quad d_2''' \quad 0 \quad c_2'''$

$0 \quad d_6''' \quad 0$

$a_{10}''' \quad 0 \quad d_{10}''' \quad 0$

$\Downarrow$

$0 \quad d_2''' \quad c_2'''$

$a_{10}''' \quad d_{10}'''$

d_6'''

$d_1' \quad c_1'$

$a_3' \quad d_3' \quad c_3'$

$a_5' \quad d_5' \quad c_5'$

$a_7' \quad d_7' \quad c_7'$

$a_9' \quad d_9' \quad c_9'$

$a_{11}' \quad d_{11}'$

$\rightarrow$

$d_1'' \quad 0 \quad c_1''$

$0 \quad 0 \quad d_3'' \quad 0 \quad c_3''$

$a_5'' \quad 0 \quad d_5'' \quad 0 \quad c_5''$

$a_7'' \quad 0 \quad d_7'' \quad 0 \quad c_7''$

$a_9'' \quad 0 \quad d_9'' \quad 0$

$a_{11}'' \quad 0 \quad d_{11}''$

$\rightarrow$

$d_1'' \quad c_1''$

$a_5'' \quad d_5'' \quad c_5''$

$a_9'' \quad d_9''$

$d_3'' \quad c_3''$

$a_7'' \quad d_7'' \quad c_7''$

$\rightarrow$

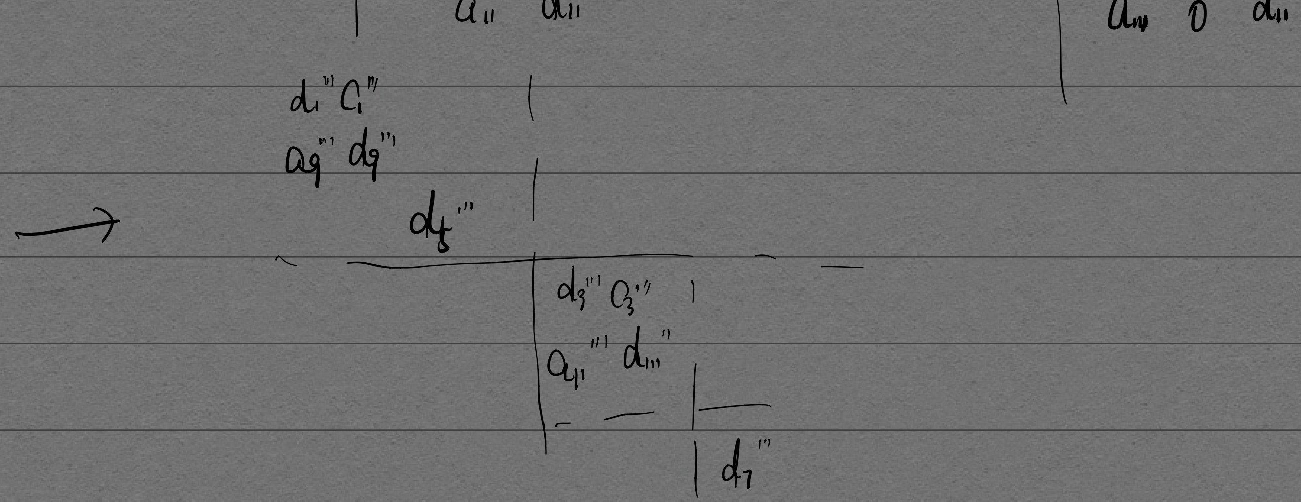
$d_1''' \quad 0 \quad c_1'''$

$0 \quad d_5''' \quad 0$

$a_9''' \quad 0 \quad d_9'''$

$d_3''' \quad 0 \quad c_3''$

$0 \quad d_7'' \quad 0$



当矩阵为 7×7 时的计算过程

每个 p 对应 (a_i, b_i, C_i, d_i)

	p0	p1	p2	p3	p4	p5	p6	p7	p8
第0次		0 1 2	1 2 3	2 3 4	3 4 5	4 5 6	5 6 7	6 7 8	
第一次	⓪ p0	① 0 1 3	② 0 2 4	③ 1 3 5	④ 2 4 6	⑤ 3 5 7	⑥ 4 6 8	⑦ 5 7 8	⑧ p8
第二次	⓪ p0	① 0 1 5	② 0 2 6	③ 0 3 7	④ 0 4 8	⑤ 1 5 8	⑥ 2 6 8	⑦ 3 7 8	⑧ p8

线程的间隔为 $2^1=2$，上一步操作的间隔为 $2^{1-1}=1$

当矩阵为 11×11 时的计算过程

第0次	p0	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	p12
$2^{1-1}=1$ 第1次'		(0,1,2)	(1,2,3)	(2,3,4)	(3,4,5)	(4,5,6)	(5,6,7)	(6,7,8)	(7,8,9)	(8,9,10)	(9,10,11)	(10,11,12)	
$2^{2-1}=2$ 第1次''		(0,1,3)	(0,2,4)	(1,3,5)	(2,4,6)	(3,5,7)	(4,6,8)	(5,7,9)	(6,8,10)	(7,9,11)	(8,10,12)	(9,11,12)	
$2^{3-1}=4$ 第2次'''		(0,1,5)	(0,2,6)	(0,3,7)	(0,4,8)	(1,5,9)	(2,6,10)	(3,7,11)	(4,8,12)	(5,9,12)	(6,10,12)	(7,11,12)	
$2^{4-1}=8$ 第3次''''		(0,1,9)	(0,2,10)	(0,3,11)	(0,4,12)	(0,5,12)	(0,6,12)	(0,7,12)	(0,8,12)	(1,9,12)	(2,10,12)	(3,11,12)	

idx_row = 4 5 6 stride = 8

```
P = (d/2 + (d%2)) * (tidx % 2) + (int) tidx/2;
```

※ r-1 中的 (2,4,6) 代表着可以通过 2和6行将 4行中的系数消成 0和8列，即 (0,4,8)

※ 这里面的数字代表构成每行系数所在的列数

1 7
2 8
3 9
4 10
5 11
6

b_i C_i d_i

a_k b_k d_k

row_idx = 3

3 − 16 = −13

$-13+8=-5$

$-5+8=3$

$3+8=11$

第1种情况 $i-stride$ i $i+stride$ $j-stride$ j $j+stride$

$i = j - stride$

$i + stride = j$

条件 $\begin{cases} i-stride < 0 \\ j+stride \geq Dmax \end{cases}$

$\Rightarrow \begin{cases} i < stride \\ j < 2stride \\ j \geq Dmax - stride \\ i \geq Dmax - 2stride \end{cases}$

For i:

int pos = $i - 2\times stride$

accum = 0

iter = 0 pos < 0 accum = 0

iter = 1 pos = $i - stride < 0$ accum = 0

iter = 2 pos = i accum = 1

iter = 3 pos = $i + stride < Dmax$ accum = 2

iter = 4 pos = $i + 2stride \geq Dmax$ accum = 2

第2种情况 $i-stride$ i $i+stride$

$i-stride < 0$

$i+stride \geq Dmax$

int pos = $i - 2\times stride$

accum = 1 ⇒ 说明满足条件

第3种情况 $\underset{\geq 0}{i-stride}$ i $\underset{<Dmax}{i+stride}$ $\underset{>0}{j-stride}$ j $\underset{<Dmax}{j+stride}$

$0 \leq i = j - stride$

$i + stride = j < Dmax$

$i - stride \geq 0$ or $j + stride < Dmax$

If $i-stride \geq 0$ ⇒ $i \geq stride \rightarrow i - stride \geq 0$

$j \geq 2stride \rightarrow j - 2stride \geq 0$

For i : int pos = $i - 2\times stride$

accum = 0

iter = 1 $i - stride \geq 0$ accum ++ = 1

iter = 2 i accum++ = 2

iter = 3 $0 < i + stride < Dmax$ accum ++ = 3

因此 accum 一定 ≥ 3

For j : int pos = $j - 2\times stride$

iter = 0 accum = 0 $j - 2\times stride \geq 0$ accum = 1

iter = 1 $j - stride \geq 0$ accum = 2

iter = 2 $j > 0$ accum = 3

因此 accum 一定 ≥ 3

1011	11
101,1	5
10,11	2
1,011	1
0,1011	0

bin = 11

count_iter (int bin) {

① m = 1
bin = 5
count = 1

② m = 2
bin = 2
count = 2

③ m = 2
bin = 1
count = 3

④ m = 3
bin = 0
count = 4

count-- = 3

行 i 列

$a_i \quad b_i \quad c_i$

$2^{n+1} \qquad 2^{n+1}$

$i - 2^{n+1} < 0 \Rightarrow D_{max} \leq 2^n$

$i + 2^n \geq D_{max} \Rightarrow 2^n \geq D_{max}$

$2^{n+1} \geq 11$

iteration	stride
0	2
1	4
⋮	
n	2^{n+1}

count_iter (int bin) { bin = 8

① m = 0
bin = 4
count = 1

② m = 0
bin = 2
count = 2

③ m = 0
bin = 1
count = 3

④ m = 1
bin = 0
count = 4

1000	8
100,0	4
10,00	2
1,000	1
0,1000	0

只有 $N = 2^n$ 时，才有 m=1 的情况

101	7
10,1	2
1,01	1
0,101	0

Householder Method

Householder matrix P: $P = 1 - 2w \cdot w^T$ w real vector with $|w|^2 = 1$

$a \cdot b^T$: matrix product $\qquad a^T \cdot b$: scalar product

$$P^2 = (1 - 2w \cdot w^T) \cdot (1 - 2w \cdot w^T) = 1$$

$$P = P^{-1}, \quad P^T = P, \quad P^T = P^{-1}$$

$$P = 1 - \frac{u \cdot u^T}{H} \qquad H = \frac{1}{2}|u|^2$$

$$P^T = 1 - 2w\, w^T = P$$

$$P^2 = 1 - \frac{2\, u \cdot u^T}{H} + \frac{1}{H^2} u \cdot \underbrace{(u^T u)}_{2H} \cdot u^T = 1$$

x 由 A 的第一列构成

P acts on a given

$u = x \mp |x| e_1$

$u = x \pm |x| e_1$

$p \cdot x = \pm |x| e_1$

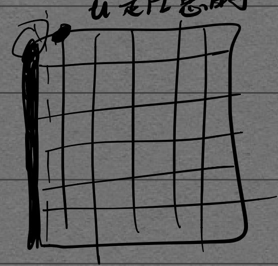
$u^2 = (x \pm |x| e_1)^T (x \pm |x| e_1) = |x|^2 + |x| x^T e_1 + |x| e_1^T x + |x|^2$

$x^T \pm |x| e_1^T$

$= 2|x|^2 + 2|x| x_1$

$h = \frac{u^2}{2} = |x|^2 + |x| x_1$

u是任意的



$A^T = A = A^{-1}$

$B^T = B = B^{-1}$

$(AB)^T = B^T A^T$

$(AB)^{-1} = B^{-1} A^{-1} = B^T A^T$

$$P_1 \cdot A = \left[\begin{array}{c|c} I & 0 \\ \hline 0 & {}^{(n-1)}P_1 \end{array}\right] \left[\begin{array}{c|c} a_{11} & A_{12} \\ \hline A_{21} & A_{22} \end{array}\right]$$

→ A_{21}与A_{12}对称

${}^{(n-1)}P_1$ 由A_{21}的第一列构成

${}^{(n-1)}P_1 \times A_{21} = \begin{bmatrix} k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ → $\pm x_{11}^{A_{21}}$ → A_{21}的第一列的第一个元素

$$\Rightarrow \left[\begin{array}{c|c} a_{11} & A_{12} \\ \hline k & \\ 0 & {}^{(n-1)}P_1 \times A_{22} \\ \vdots & \\ 0 & \end{array}\right] \times \left[\begin{array}{c|c} I & 0 \\ \hline 0 & {}^{(n-1)}P_1 \end{array}\right]$$

${}^{(n-1)}P_1 \times A_{21}$ ⇐

$$\Rightarrow \left[\begin{array}{c|cccccc} a_{11} & k & 0 & \cdots & 0 & 0 & \cdots 0 \\ \hline k & & & & & & \\ 0 & & & {}^{(n-1)}P_1 \times A_{22} \times {}^{(n-1)}P_1 = A' & & & \\ 0 & & & & & & \\ \vdots & & & & & & \\ 0 & & & & & & \end{array}\right]$$

$= A_{12} \times {}^{(n-1)}P_1 = \left({}^{(n-1)}P_1 \times A_{21}\right)^T$

A u

b_{n-1} k 0

k b_n 0

构成

2x2

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & & & & \\ \vdots & \vdots & & {}^{(n-2)}P_2 & & \\ 0 & 0 & & & & \end{bmatrix}$$

计算顺序：6, u, H, p, k, q, A'

① 6

上一步计算出来的 ← A (n×n) 对称

② $P = \frac{A u}{H}$ ① (n×1)

$A - B = A - \left(\frac{u \cdot u^T}{\ }\right) = A - p \cdot u^T$ (n×1) (1×n)

$$A' = PAP = PA - p\cdot p\cdot u^T$$

$$= \left(1 - \frac{u\cdot u^T}{H}\right)\left(A - p\cdot u^T\right)$$

$$= A - p\cdot u^T - \frac{u\cdot u^T A}{H} + \frac{u\cdot u^T p\cdot u^T}{H}$$ scalar product

$$= A - p\cdot u^T - u\cdot(Au)^T + 2K\, u u^T$$

$(Au)^T \to p^T$, $\quad 2K \to \frac{u^T p}{2H}$

① $q = p - Ku$ ②

$$= A - p\cdot u^T - u p^T + 2K u u^T$$

从n行开始 ※ $A' = A - q\cdot u^T - u\cdot q^T$

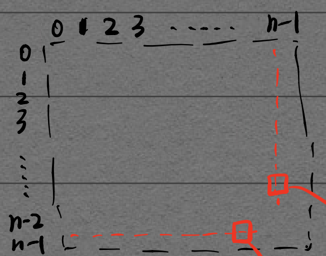
At stage $m=1$, $i = n-m+1 = n$. $u^T = [a_{n1}, a_{n2}, a_{n3}, a_{n4}, \ldots\ldots, a_{n,n-1} \pm \sqrt{6}, 0]$ ⇒ 因为A是对称的

$m=2$, $i = n-m+1 = n-1$ $u^T = [a_{n-1,1}, a_{n-1,2}, a_{n-1,3}, a_{n-1,4} \cdots, a_{n-1,n-2} \pm\sqrt{6}, 0, 0]$

tred_fact_k ( ):

$h = 6$, $g = \sqrt{6} = |x|$

sAds[nt + i*n + k] $\quad l = i-1$, $k \le l$



If $i = n-1$,

$\underbrace{nt + i*n}_{i行} + \underbrace{k}_{k列}$ 取得是 $(n-1, 0)$, $(n-1, 1)$, $(n-1, 2)$, $(n-1, 3)$, …, $(n-1, n-2)$

sAds[nt + i*n + k] 是 A的最后一列的前 $i-1$ 个元素.

f 最后一个元素, 用来计算次对角元素 k.

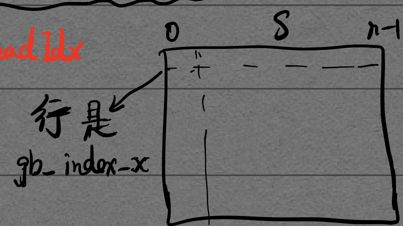
[nt + i*n + l] → $l = i-1$

$h = 6 = (a_{i1})^2 + \cdots + (a_{i,i-1})^2$.

$g = \pm\sqrt{6}$, ±号 取决于 f.

$\underbrace{\text{S[gb\_index\_x * n + i]}}_{}$ = scale * g; S保存次对角的k值

第i号 threadIdx



S从 1到 n-1

$h = \frac{1}{2}|u|^2$

$u = x \mp |x| e_1$

⇐ $\underbrace{h \mathrel{-}= f*g}$ ⟶ $u = x \mp |x| e_1$ ($|x| \to g$)

计算 u中变化的那个值.

$h = \frac{u^2}{2} = x^2 \pm |x| x_1$

$x^2 \to 6$, $|x| \to g$, $x_1 \to f$

sAds[nt + i*n + l] = f − g // store u

$$A' = P_1 A P_1$$

$$A'' = P_2 A' P_2 = P_2 P_1 A P_1 P_2$$

$tridiag\_A = Q'AQ$ $\quad Q\ tridiag\_A\ Q' = A$ $\quad Ax = b$

$Q \cdot tA \cdot Q'x = b$

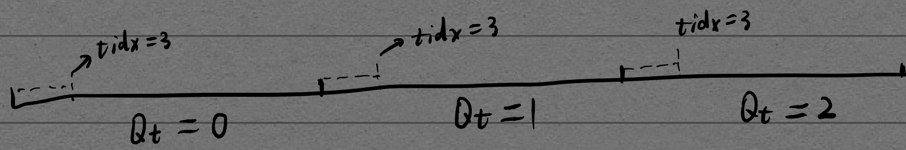
$tA\,(Q'x) = Q'b \Rightarrow Q'x = y \Rightarrow x = Qy$

q sy

PCR — Parallel cyclic reduction

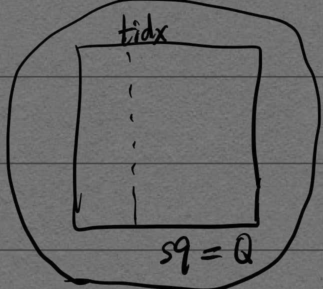
一个block里

```
tidx = threadIdx.x % n;
Qt = (threadIdx.x - tidx)/n;
gb_index_x = Qt + blockIdx.x * (blockDim.x/n);
```

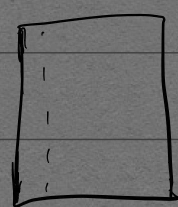


blockDim.x = 3×N

最简单情况: blockDim.x = N $\quad$ Qt=0, gb_index_x = 0



要计算的是 $Q'b$



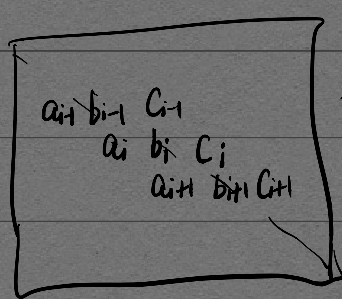
```
for i=0; i<n; i++ {
    sq[i*n + tidx]          → sq的 tidx列
    sum += sq[i*n + tidx] * sy[i];    → sy的第i个元素
}
__syncthreads();
sy[tidx] = sum;
```

tidx 是中间的 i行, tL = i-1行, tR = i+1行.

如果 tL, tR 超出范围, 为0.

迭代 for:

tL = 是



因为对称, 所以

$a_i = c_{i-1}$