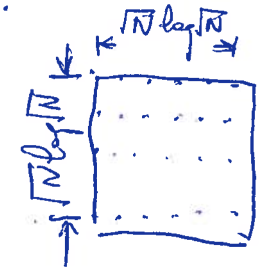
Following Ideas are for 2D, but it can easily be ~~extended to~~ d-dimension space

(Page 1)

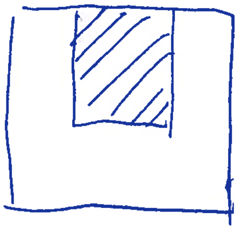
① For Pure dimension reduction algor. ~~assuming~~ it adopts Yao's 1D algor.

	space/~~time~~ for preprocess	query time
$k=0$	N^2	0
$k=1$	$N\log^2 N$	$(2k-1)^2$
$k=2$	$N(\log^* N)^2$	$(2k-1)^2$
$k=3$	$N(\log^{**} N)^2$	$(2k-1)^2$
⋮	⋮	⋮
~~k~~ $k=\alpha(N)$	$N\alpha^2(N)$	$\alpha^2(N)$



Proof: basic idea. 2D array is a ~~1D~~ 1D array of 1D array.
For each horizontal array, we call Yao's algo. to get a $\Theta(\sqrt{N}\log\sqrt{N})$ complexity data structure, then call the same algor. on vertical 1D array. ~~At this~~ time, for the vertical 1D array, each element of which is a $\Theta(\sqrt{N}\log\sqrt{N})$ data structure. so ~~#~~ in total, the complexity is $\Theta(\sqrt{N}\cdot\log\sqrt{N}(\sqrt{N}\log\sqrt{N}))$
$\boxed{=\Theta(N\log^2 N)}$

② For query type that touches one boundary. we only requires a preprocess algor of complexity $\Theta(N\log N)$, $\Theta(N\log^* N)$ ... to handle them in query time $(2k-1)$; ... $\alpha(N)$. (Lemma 1)



Idea: for each vertical 1D array, we store the ~~partial~~ partial sum counted from top. ⇒ it becomes a 1D <horizontal> array of 1D array <vertical>

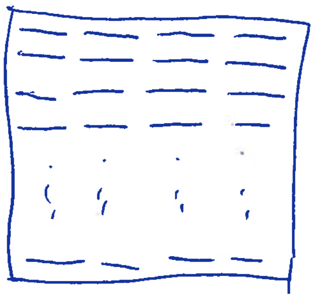
⇒ partial sum counted from the top

~~the element of each~~
each element of horizontal array is of size $\sqrt{N}$
⇒ In total $(\sqrt{N}\cdot\log\sqrt{N})\cdot\sqrt{N} = \underline{N\log N}$

(page 2)

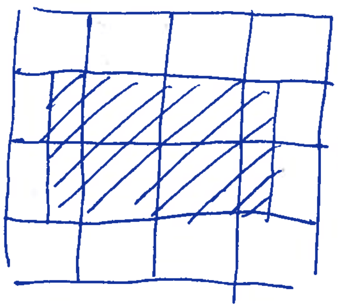
In data reduction method, we only reduce the data ~~according~~ aligned in line segment. each line segment is of length w. So by one level of data reduction it will have G/w points. if input algor. is S & S', (S to handle

G all possible types of query, S' to handle all query type that touches one side of boundary),

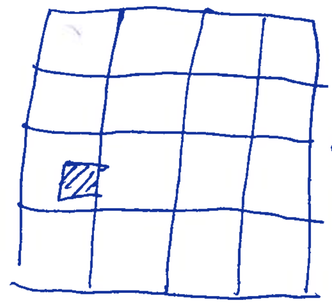


$\rightarrow| w |\leftarrow$

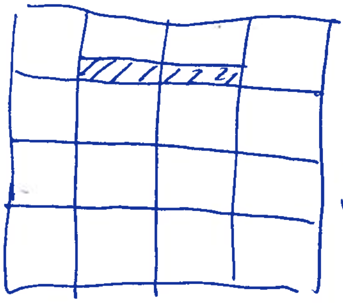
$$\Rightarrow \underbrace{2S\left(\frac{G}{w}\right)}_{\text{type 1}} + \underbrace{\frac{G}{w^2}\cdot S'(w^2)}_{\text{type 2}} + \underbrace{\frac{2\sqrt{G}}{w}\cdot S'\left(w\left(\sqrt{G}/w\right)\right)}_{\text{type 3}} + \underbrace{4G}_{\text{type 4}}$$



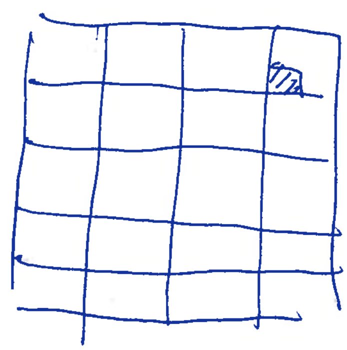
type 1 query, which can be answered in $q(k-1,d)$.



Type 2 query; which can be answered in $\sqrt{q(k-1,d)}$ or $\sqrt{q(k,d)}$



~~type~~ 3 query, which can be answered in $\sqrt{q(k-1,d)}$ or $\sqrt{q(k,d)}$



Type 4 query, which can be answered in $O(L)$.

---

by 2-level of data blocking. it requires

$$2S\left(\frac{G}{w}\right) + \frac{G}{w^2}S'(w^2) + \frac{2\sqrt{G}}{w}\cdot S'\left(w\cdot\frac{\sqrt{G}}{w}\right) + 4G$$

$$+ \frac{G}{w^2}\left\{\left(2\cdot S\left(\frac{w^2}{\sqrt{w}}\right) + \frac{w^2}{w}S'(\sqrt{w}\cdot\sqrt{w}) + \frac{2w}{\sqrt{w}}S'\left(\sqrt{w}\cdot\frac{w}{\sqrt{w}}\right)\right\} + 4G\right)$$

$$= 2\cdot S\left(\frac{G}{w}\right) + \frac{G}{w^2}\cdot S'(w\cdot w) + \frac{2\sqrt{G}}{w}\cdot S'\left(w\cdot\frac{\sqrt{G}}{w}\right) +$$

$$\frac{G}{w^2}\left\{\left(2\cdot S(w^{\frac{3}{2}}) + w\cdot S'(\sqrt{w}\cdot\sqrt{w}) + 2\cdot w^{\frac{1}{2}}\cdot S'\left(\sqrt{w}\cdot\cancel{\frac{w}{\sqrt{w}}}\sqrt{w}\right)\right)\right\} + O(G)$$

Suppose input $\{S, G+\}$. $S = N\log^2 N$, $S' = N\log N$. $G = N$. $w = \log N$. $q(k,d)$. (Page 3)

$\Rightarrow$ Entire Complexity has recurrence,

$$T(N) = 2S\left(\frac{G}{w}\right) + \frac{G}{w^2} S'(w \cdot w) + \frac{2\sqrt{G}}{w} \cdot S'\left(w \cdot \frac{\sqrt{G}}{w}\right)$$

$$+ \frac{G}{w^2}\left\{\left(2 \cdot S(w^{\frac{3}{2}}) + w \cdot S'(\sqrt{w} \cdot \sqrt{w}) + 2w^{\frac{1}{2}} \cdot S'(\sqrt{w} \cdot \sqrt{w})\right)\right\} + \frac{N}{\log N} T(\log N)$$

$$= 2 \cdot S\left(\frac{N}{\log N}\right) + \frac{N}{\log^2 N} S'(\log N \cdot \log N) + \frac{2\sqrt{N}}{\log N} \cdot S'\left(\log N \cdot \frac{\sqrt{N}}{\log N}\right)$$

$$+ \frac{N}{\log^2 N}\left\{2 \cdot S(\log^{\frac{3}{2}} N) + \log N \cdot S'(\sqrt{\log N} \cdot \sqrt{\log N}) + 2\sqrt{\log N} \cdot S'(\sqrt{\log N} \cdot \sqrt{\log N})\right\} + \frac{N}{\log N} T(\log N)$$

$$= \underline{2 \cdot \frac{N}{\log N}(\log N - \log\log N)^2} + \frac{N}{\cancel{\log^2 N}} \cdot \cancel{\log^2 N} \cdot \log\log N + \frac{2\sqrt{N}}{\cancel{\log N}} \cdot \sqrt{N} \cdot \cancel{\log}\left(\frac{\sqrt{N}}{\cancel{\log N}}\right)$$

$$+ \frac{N}{\log^2 N}\left\{2 \cdot \log^{\frac{3}{2}} N \left(\frac{3}{2}\right)^2 \cdot \log^2\log N + \underline{\log N \cdot \log N} \cdot \frac{1}{2}\log\log N + 2\sqrt{\log N} \cdot \log N \cdot \frac{1}{2}\log\log N\right\}$$

$$+ \frac{N}{\log N} T(\log N)$$

$$= O(N \cdot \log N) + O(N) + \frac{N}{\log N} T(\log N) = O(N \log N \log^* N).$$

$$q(k+1, d) = \underbrace{q(k,d)}_{\substack{\text{at most 1} \\ \text{type 1 query}}} + \underbrace{2\sqrt{q(k,d)}}_{\substack{\text{at most 2} \\ \text{~~type~~ 2 or} \\ \text{type 3 query}}} + \underbrace{O(1)}_{\substack{\text{type 4} \\ \text{query}}}.$$

Recall that $q(k,d) = (2k-1)^2$. $\Rightarrow \sqrt{q(k,d)} = 2k-1$.

$\Rightarrow q(k+1, d) = (2k-1)^2 + 2(2k-1) + O(1) \doteq \underline{4k^2 + O(1)}$.

Now $S = N\log N \log^* N$. select $S' = N\log^* N \cdot (g(k+1, d))$ $G = N$. $w = \log N$

$$\Rightarrow T(N) = 2S(\frac{G}{w}) + \frac{G}{w^2} S'(w \cdot w) + \frac{2\sqrt{G}}{w} \cdot S'(w \cdot \frac{\sqrt{G}}{w})$$ (Page 4)

$$+ \frac{G}{w^2}\{(2 \cdot S(w^{\frac{3}{2}}) + w S'(\sqrt{w} \cdot \sqrt{w}) + 2w^{\frac{1}{2}} S'(\sqrt{w} \cdot \sqrt{w})\} + \frac{N}{\log N} T(\log N)$$

$$= 2S(\frac{N}{\log N}) + \frac{N}{\log^2 N} S'(\log N \cdot \log N) + \frac{2\sqrt{N}}{\log N} \cdot S'(\log N \cdot \frac{\sqrt{N}}{\log N})$$

$$+ \frac{N}{\log^2 N}\{2 \cdot S(\log^{\frac{3}{2}} N) + \log N \cdot S'(\sqrt{\log N} \cdot \sqrt{\log N}) + 2\log^{\frac{1}{2}} N \cdot S'(\sqrt{\log N}\sqrt{\log N})\}$$

$$+ \frac{N}{\log N} T(\log N)$$

$$= 2\frac{N}{\log N} \cdot (\log N - \log\log N) \log^*(\frac{N}{\log N}) + \frac{N}{\log^2 N} \log^2 N \cdot \log^*(\log N)$$

$$+ \frac{2\sqrt{N}}{\log N} \cdot \sqrt{N} \cdot \log^*(\frac{\sqrt{N}}{\log N}) + \frac{N}{\log^2 N}\{2 \cdot \log^{\frac{3}{2}} N \cdot \frac{3}{2}\log\log N \cdot \log^*(\log^{\frac{3}{2}} N)$$

$$+ \underline{\log N \cdot \log N} \cdot \log^*(\sqrt{\log N}) + 2\log^{\frac{1}{2}} N \cdot \log N \cdot \log^* \sqrt{\log N}\}\} + \frac{N}{\log N} T(\log N)$$

$$= N\log^* N + \frac{N}{\log N} T(\log N)$$

$$\Rightarrow T(N) = N(\log^* N)^2.$$

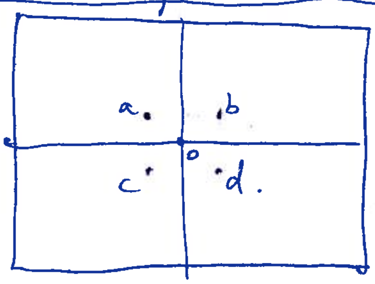
$$\underline{g(k+2, d)} = g(k+1, d) + 2\sqrt{g(k+1, d)} + O(1).$$
$$= 4k^2 + 2(2k+1) + O(1)$$
$$= 4k^2 + 4k + O(1) = \underline{(2k+1)^2}.$$

Keep doing above procedure, it's easy to prove by reduction, we have exactly the same bound in preprocessing time (space and query as in Dimension Reduction Method.

Suppose the input $S = N \log N \xrightarrow{\text{select}} S' = N \log^* N$

by above procedure, we can easily get ~~that~~

$\underset{\uparrow \text{ meta-algor.}}{M}(S) = N \log^* N$. $\qquad \underset{\uparrow \text{ meta-algo.}}{m}(q) = \underline{4k + \cancel{k}\, O(1)}$



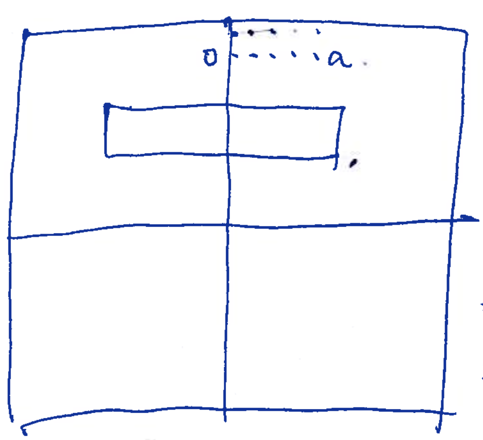
Now, we design the input algor. as follows.

(1) Divide the entire region into 4 parts.

(2) for points in each part, store it's partial sum relative to point o. (the center point).

(3) Apparently, by keep doing recursion, we use $N \log N$ space & time to handle all queries that spans all 4 parts in $\underline{4}$ (ops).

(4) the left-over ~~is~~ the long strip query which spans only 2 parts. (if completely within one part, we handle it by recursion.)



For long strip query, we adopt the same trick as in Lemma 1. for each point, such as a in diagram, store the partial sum relative to the left point on the same horizontal line (o).

⇒ reduce it to 1D problem.

⇒ select the 1D algor. ~~as~~ of complexity $N \log^* N$.

⇒ total complexity to handle long strip query at this level is $(N \log N \log^* N)$.

query time is $\underline{3}$ (op).

Now we have the input algor. of complexity $N \log N \log^* N$, query complexity 3 (op), by doing one level of data reduction,
such as the procedure on Page 4. we get an algor of complexity $\Rightarrow N (\log^* N)^2$, but query complexity, $3+4=7$.

conjecture the ~~query~~ complexity is $\underline{4k+3}$ in 2D, instead of $(2k-1)^2$