

1. Deal $\begin{cases} d_i \leftarrow d_i + x \quad (i=l, \dots, r) \\ \text{get } d_l + \dots + d_r \end{cases}$
 queries without lazy propagation

* Use Fenwick Tree (faster)

① We'll handle on prefix sum array ($= S[1, \dots, n]$)

② We'll handle 2 values on each index (a_i, b_i)

③ $S_i := d_l + \dots + d_i := (a_l + \dots + a_i) \cdot i + (b_l + \dots + b_i)$

④ Let's consider update query

$d_i \leftarrow d_i + x \quad (i=l, \dots, r)$

$\Leftrightarrow S_i \leftarrow S_i + x(i-l+1) \quad (i=l, \dots, r)$

$S_i \leftarrow S_i + x(r-l+1) \quad (i=r+1, \dots, n)$

$\Leftrightarrow \begin{cases} a_l \leftarrow a_l + x \\ b_l \leftarrow b_l - x(l-1) \\ a_{r+1} \leftarrow a_{r+1} - x \\ b_{r+1} \leftarrow b_{r+1} + xr \end{cases} \rightarrow \text{Use Fenwick}$

⑤ Let's consider sum query

$S_r = \underbrace{(a_l + \dots + a_r)}_{\text{get by Fenwick}} \cdot r + \underbrace{(b_l + \dots + b_r)}_{\text{get by Fenwick}}$

$d_l + \dots + d_r = S_r - S_{l-1}$