The main idea behind the solution of this problem is simplifying the equation. For a query operation we need the answer of this equation-
1A_l + (1+d)A_l+1 + (1+2d)A_l+2 + (1+3d)A_l+3 + … + (1+(r-l)d)A_r
Now let the query range l r is 1 3. and the given array is A=[a1,a2,a3,a4,…]

So our quation will be-
1*a1 + (1+d)*a2 + (1+2d)*a3 + (1+3d)*a4
= a1 + a2 + d*a2 + a3 + 2d*a3 + a4 + 3d*a4
= (a1+a2+a3+a4) + d*(a2+ 2*a3 + 3*a4)

Now We can see that first part of this equation is only sum query which can be done using a segment tree and for 2nd part we can use a trick. We can pree calculate the sequence like this-
1*a1 + 2*a2 + 3*a3 + 4*a4 + 5*a5 + 6*a6 + 7*a7 + ……
When we need the 2nd part of the equation we can use this equation. We can get the 2nd part of the equation from this equation by this way-
(2*a2 + 3*a3 + 4*a4)-(a2+a3+a4) = (a2+ 2*a3 + 3*a4).

Now the above equation is also only a sum equation. We can apply any range sum query or range sum update on this equation and the the first sum equation.

So if we store this two equation in two segment tree then we can perform range update or query on the segment tree and get any range sum query from the segment tree and using the equation we can answer each query of the problem correctly.