RANGE QUERIES

Αδάμος Ττοφαρή

Περιεχόμενα

- Εισαγωγή
- Ερωτήματα σε στατικούς πίνακες
  - Ερωτήματα Αθροίσματος
  - Ερωτήματα Ελάχιστης τιμής
- Binary Index Trees
  - Δομή
  - Ερωτήματα
  - Ενημερώσεις

Εισαγωγή

Θα συζητήσουμε τις δομές δεδομένων που μας αφήνουν να επεξεργαστούμε αποδοτικά ερωτήματα διαστημάτων. Όταν πρέπει να απαντήσουμε ένα ερώτημα διαστημάτων, πρέπει να υπολογίσουμε την απάντηση του υποπίνακα ενός πίνακα. Τυπικές ερωτήσεις διαστημάτων είναι:

- Sum(a, b): υπολογίζει το άθροισμα των τιμών στο διάστημα [a,b]
- RMQ(a, b): υπολογίζει την ελάχιστη τιμή στο διάστημα [a,b]

Για παράδειγμα, σκεφτείτε το διάστημα [3,6] στον επόμενο πίνακα:

0	1	2	3	4	5	6	7
1	3	8	4	6	1	3	4

Σε αυτή την περίπτωση, sum(3,6)=14, min(3,6)=1, max(3,6)=6

Μια απλή μέθοδος να υπολογιστούν τα ερωτήματα είναι να διασχίσουμε τις τιμές του διαστήματος με ένα for.

```
int sum(int a, int b) {
    int s = 0;
    for (int i = a; i <= b; i++) {
        s += array[i];
    }
    return s;
}
```

Αυτή η συνάρτηση δουλεύει σε χρόνο O(n), οπού n το μέγεθος του πίνακα. Έτσι μπορούμε να υπολογίσουμε q ερωτήματα σε χρόνο O(nq). (**Time limit Exceeded**)

Ερωτήματα σε στατικούς πίνακες

Αρχικά θα μελετήσουμε την περίπτωση που οι πίνακες είναι στατικοί, δηλαδή οι τιμές του δεν αλλάζουν κατά την διάρκεια των ερωτημάτων.

Ερωτήματα Αθροίσματος

Μπορούμε να επεξεργαστούμε ευκολά τα ερωτήματα αφού δημιουργήσουμε πίνακα προθεμάτων αθροισμάτων. Κάθε τιμή στον πίνακα θα αποθηκεύει το άθροισμα όλων των τιμών μέχρι την συγκεκριμένη θέση. Δηλαδή η τιμή στην θέση k θα είναι ίση με sum(0, k). Ο πίνακας μπορεί να υπολογιστεί σε χρόνο O(n).

Για παράδειγμα υποθέστε τον επόμενο πίνακα:

0	1	2	3	4	5	6	7
1	3	4	8	6	1	4	2

Ο αντίστοιχος πίνακας προθεμάτων θα είναι ο επόμενος:

0	1	2	3	4	5	6	7
1	4	8	16	22	23	27	29

Αφού γνωρίζουμε όλες τις τιμές sum(0, k) , μπορούμε να υπολογίσουμε κάθε τιμή του sum(a, b) σε O(1) χρόνο:

$$sum(a,b) = sum(0,b) - sum(0,a-1)$$

Οπού sum(0, -1)=0 για να ισχύει ο τύπος όταν a=0.

Για παράδειγμα, σκεφτείτε το διάστημα [3, 6]:

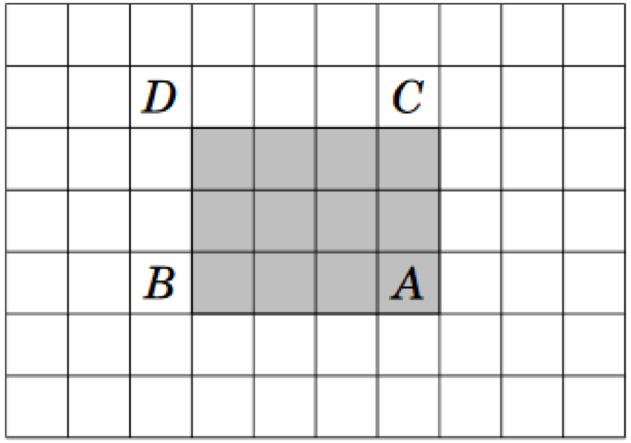
0	1	2	3	4	5	6	7
1	3	4	8	6	1	4	2

Σε αυτή την περίπτωση sum(3,6)=8+6+1+4=19. Αυτό το άθροισμα μπορεί να υπολογιστεί από δυο τιμές στον πίνακα προθεμάτων

0	1	2	3	4	5	6	7
1	4	8	16	22	23	27	29

Έτσι το άθροισμα είναι sum(3,6)=sum(0,6)-sum(0,2)=27-8=19.

Είναι επίσης δυνατό να γενικεύσουμε την ιδέα για περισσότερες διαστάσεις. Για παράδειγμα μπορούμε να κάνουμε ένα δισδιάστατο πίνακα προθεμάτων ο οποίος μπορεί να υπολογίζει άθροισμα σε O(1) χρόνο. Κάθε θέση αντιστοιχεί στο άθροισμα όλων τιμών από την πάνω αριστερή γωνιά.



Το άθροισμα μπορεί να υπολογιστεί με τον επόμενο τύπο:

$$S(A) - S(B) - S(C) + S(D)$$

Οπού S(X) αντιστοιχεί στο άθροισμα των τιμών από την πάνω αριστερή γωνιά.

Ερωτήματα Ελάχιστης τιμής

Επόμενος θα δούμε πως μπορούμε να υπολογίσουμε ελάχιστη τιμή σε χρόνο O(1) μετρά από O(nlogn) προϋπολογισμό. Σημειώστε ότι η μέγιστη τιμή θα υπολογιζόταν με τον ίδιο τρόπο.

Η ιδέα είναι να προϋπολογίσουμε όλες τις τιμές rmq(a,b) οπού b-a+1, το μήκος του διαστήματος, είναι δύναμη του δυο.

Για παράδειγμα ο πίνακας:

0	1	2	3	4	5	6	7
1	3	4	8	6	1	4	2

Θα υπολογιστούν οι επόμενες τιμές:

a	b	$\mathrm{rmq}(a,b)$
1	1	1
2	2	3
3	3	4
4	4	8
5	5	6
6	6	1
7	7	4
8	8	2

a	b	$\mathrm{rmq}(a,b)$
1	2	1
2	3	3
3	4	4
4	5	6
5	6	1
6	7	1
7	8	2

a	b	$\mathrm{rmq}(a,b)$
1	4	1
2	5	3
3	6	1
4	7	1
5	8	1
1	8	1

Το πλήθος των προϋπολογισμένων τιμών είναι O(nlogn), γιατί υπάρχουν O(logn) μήκη διαστημάτων που είναι δυνάμεις του δυο. Επίσης οι τιμές μπορούν υπολογιστούν αναδρομικά χρησιμοποιώντας τον αναδρομικό τύπο.

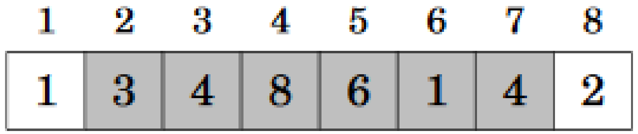
$$rmq(a,b) = \min(rmq(a, a+w-1), rmq(a+w, b))$$

Οπού b-a+1 είναι δύναμη του δυο και $w = \frac{(b-a+1)}{2}$. Ο υπολογισμός των τιμών χεριάζετε O(nlogn) χρόνο.

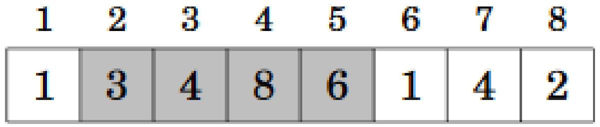
Μετά από αυτό μπορούμε να υπολογίσουμε το rmq(a,b) σε χρόνο O(1) ως την μικρότερη τιμή δυο προϋπολογισμένων αριθμών. Έστω k η μεγαλύτερη δύναμη του δυο που δεν ξεπερνά το b-a+1. Μπορούμε να υπολογίσουμε το rmq(a,b) χρησιμοποιώντας τον τύπο.

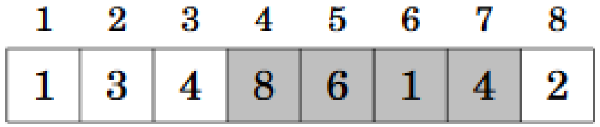
$$rmq(a,b) = \min(rmq(a, a+k-1), rmq(b-k+1, b))$$

Ως παράδειγμα, σκεφτείτε το επόμενο διάστημα [2,7].



Το μήκος του διαστήματος είναι 6, και η μεγαλύτερη δύναμη του δυο που δεν ξεπερνά το 6 είναι 4. Έτσι το διάστημα είναι η ένωση των διαστημάτων [2,5] και [4,7]:





Αφού rmq(2,5)=3 και rmq(4,7)=1, μπορούμε να συμπεράνουμε ότι rmq(2,7)=1.

Binary Index Tree

Ένα BIT ή Fenwick Tree μπορεί να παρομοιαστεί ως μια δυναμική έκδοση του πίνακα προθεμάτων. Αυτή η δομή μπορεί να εκπληρώσει δυο O(logn) διεργασίες:

- Να υπολογίζει το άθροισμα των τιμών σε ένα διάστημα
- Ενημέρωση της τιμής σε μια θέση

Το πλεονέκτημα ενός BIT είναι ότι μας επιτρέπει να ενημερώνουμε αποδοτικά τον πίνακα με τις τιμές μεταξύ των ερωτημάτων. Αυτό δεν θα ήταν δυνατό με ένα πίνακα προθεμάτων γιατί, για κάθε ενημέρωση, θα χρειαζόταν να αναδημιουργούμε τον πίνακα.

Δομή

Ένα BIT μπορεί να αναπαρασταθεί ως ένας πίνακας οπού η τιμή του στην θέση x είναι ίση με το άθροισμα των τιμών στο διάστημα [x-k+1, x], οπού k είναι ο μεγαλύτερη δύναμη του δυο που διαίρει το x. Για παράδειγμα, αν x=6, τότε k=2, γιατί το 2 δείρει το 6 αλλά το 4 όχι.

Έστω ο επόμενος πίνακας:

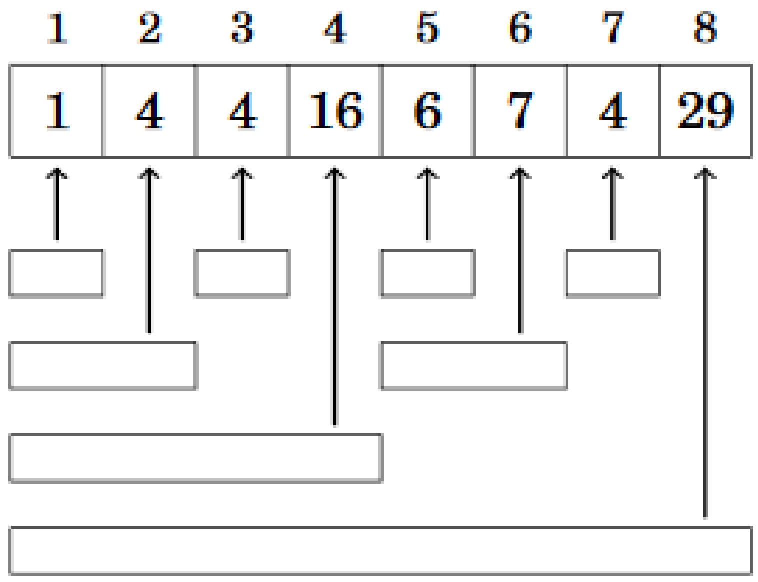
1	2	3	4	5	6	7	8
1	3	4	8	6	1	4	2

Ο αντίστοιχος BIT είναι:

1	2	3	4	5	6	7	8
1	4	4	16	6	7	4	29

Για παράδειγμα, η τιμή στη θέση 6 στο BIT είναι 7 γιατί το άθροισμα των στοιχείων στο διάστημα [5,6] είναι 6+1=7.

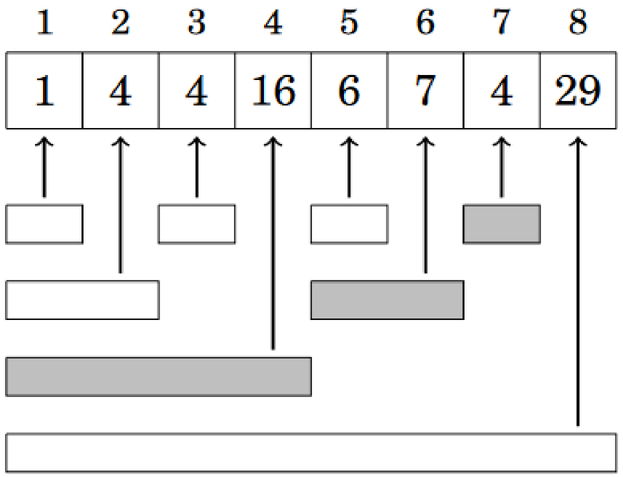
Η επόμενη εικόνα δείχνει πως κάθε τιμή του BIT αντιστοιχεί σε διάστημα στον πίνακα:



Ερωτήματα

Οι τιμές του BIT μπορούν να χρησιμοποιηθούν αποδοτικά για να υπολογίσουμε το αθροισμάτων τιμών στο διάστημα [1,k], γιατί κάθε διάστημα μπορεί να μοιραστεί σε O(logn) διαστήματα τα οποία έχουν υπάρχουν στο BIT.

Για παράδειγμα, το διάστημα [1,7] αντιστοιχεί στις επόμενες τιμές:



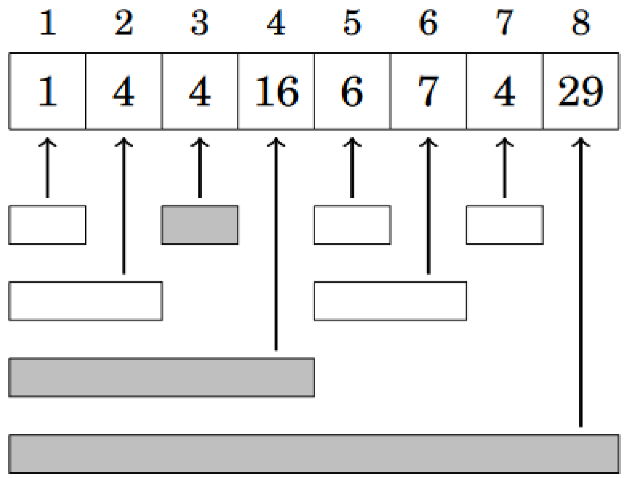
Έτσι, το άθροισμα των τιμών στο διάστημα είναι [1,7] είναι 16+7+4=27.

Για να υπολογίσουμε το άθροισμα των στοιχειών σε οποιοδήποτε διάστημα [a,b], μπορούμε να χρησιμοποιήσουμε τον τύπο.

$$sum(a, b) = sum(1, b) - sum(1, a - 1)$$

Ενημερώσεις

Όταν μια τιμή στον πίνακα ενημερώνετε, πολλαπλές τιμές στο BIT πρέπει να ενημερωθούν. Για παράδειγμα, αν ένα στοιχείο στη θέση 3 ενημερωθεί, τα αθροίσματα των επομένων διαστημάτων πρέπει να ενημερωθούν



Αφού κάθε στοιχείο ανήκει σε O(logn) διαστήματα, η ενημέρωση γίνετε σε ίδιο χρόνο

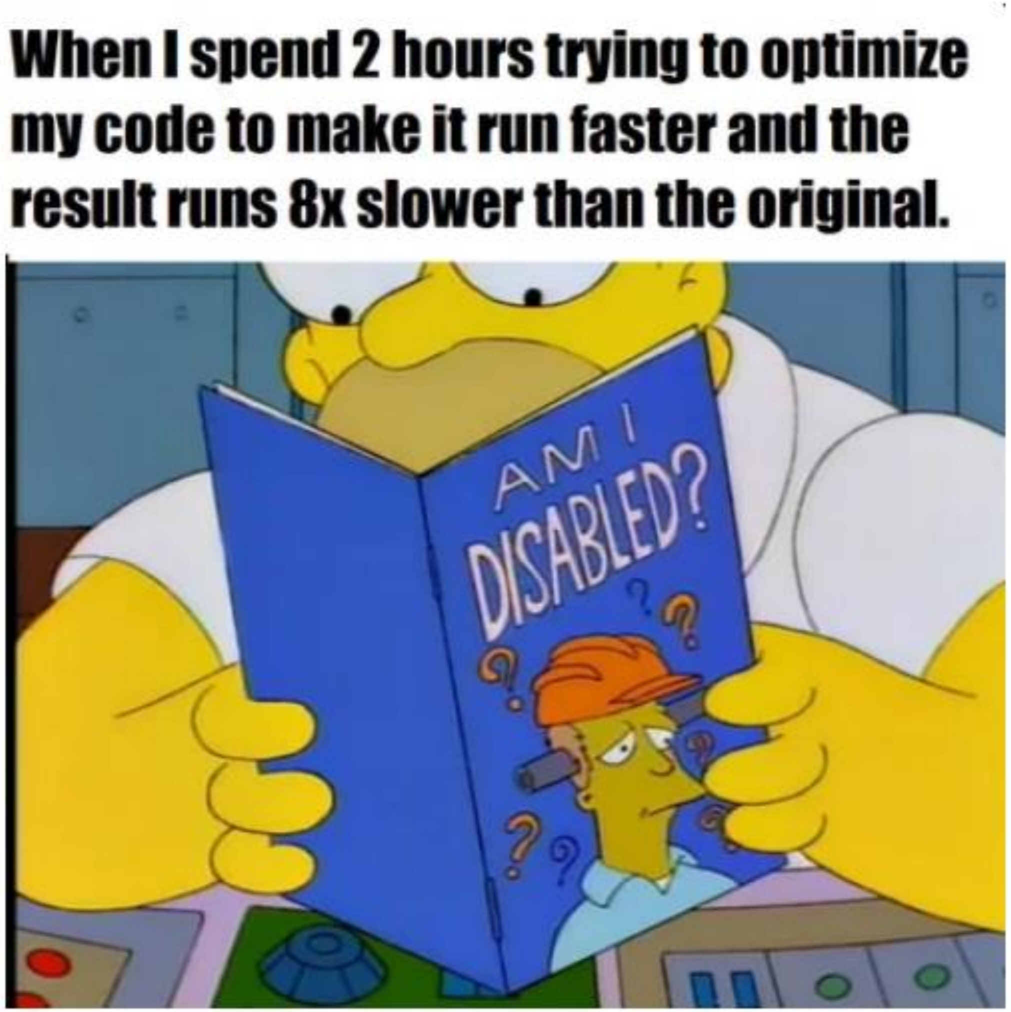
Υλοποίηση

Οι διαδικασίες στο BIT μπορούν να γίνουν με ένα έντεχνο τρόπο και αποδοτικό τρόπο χρησιμοποιώντας δυαδικούς τελεστές. Το κλειδί είναι $k \ \& \ -k$ ξεχωρίζει το τελευταίο bit του k. Για παράδειγμα, 26 & -26=2 γιατί ο αριθμός 26 αντιστοιχεί στο 11010 και το 2 στο 10.

Αποδεικνύετε ότι όταν θέλουμε να κάνουμε ερώτημα σε ένα BIT η θέση k πρέπει να μειωθεί κατά k&-k σε κάθε βήμα και για ενημέρωση η θέση k πρέπει να αυξηθούν κατά k&-k σε κάθε βήμα.

```
int sum(int k) {
    int s = 0;
    while (k >= 1) {
        s += b[k];
        k -= k&-k;
    }
    return s;
}
```

```
void add(int k, int x) {
    while (k <= n) {
        b[k] += x;
        k += k&-k;
    }
}
```



adamos2468@gmail.com