

Προηγμένα Θέματα Αλγορίθμων

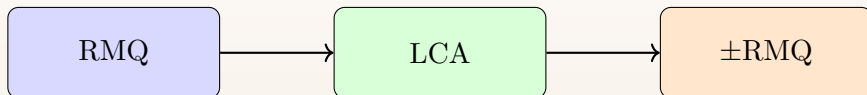
Διαλέξη 2: Δομές Δεδομένων II

Βασίλειος Νάκος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Αθηνών

Σχέση μεταξύ RMQ και LCA.

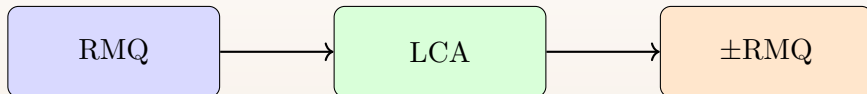
Σήμερα θα ανακαλύψουμε τη σχέση μεταξύ των RMQ, LCA.



I. $\text{RMQ} \Rightarrow \text{LCA}$: Αν υπάρχει μια δομή χρόνου $\langle f(n), g(n) \rangle$ για το LCA τότε υπάρχει μια δομή χρόνου $\langle f(n), g(n) \rangle + O(n)$ για το RMQ.

Σχέση μεταξύ RMQ και LCA.

Σήμερα θα ανακαλύψουμε τη σχέση μεταξύ των RMQ, LCA.

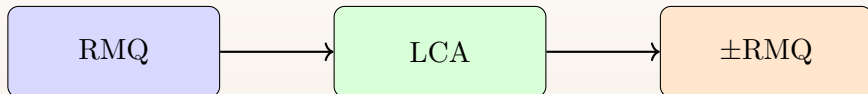


I. $\text{RMQ} \Rightarrow \text{LCA}$: Αν υπάρχει μια δομή χρόνου $\langle f(n), g(n) \rangle$ για το LCA τότε υπάρχει μια δομή χρόνου $\langle f(n), g(n) \rangle + O(n)$ για το RMQ.

II. $\text{LCA} \Rightarrow \pm\text{RMQ}$: Αν υπάρχει δομή χρόνου $\langle f(n), g(n) \rangle$ για το $\pm\text{RMQ}$ τότε υπάρχει μια δομή χρόνου $\langle f(2n), g(2n) \rangle + O(n)$ για το LCA.

Σχέση μεταξύ RMQ και LCA.

Σήμερα θα ανακαλύψουμε τη σχέση μεταξύ των RMQ, LCA.



I. $\text{RMQ} \Rightarrow \text{LCA}$: Αν υπάρχει μια δομή χρόνου $\langle f(n), g(n) \rangle$ για το LCA τότε υπάρχει μια δομή χρόνου $\langle f(n), g(n) \rangle + O(n)$ για το RMQ.

II. $\text{LCA} \Rightarrow \pm\text{RMQ}$: Αν υπάρχει δομή χρόνου $\langle f(n), g(n) \rangle$ για το $\pm\text{RMQ}$ τότε υπάρχει μια δομή χρόνου $\langle f(2n), g(2n) \rangle + O(n)$ για το LCA.

III. Θα σχεδιάσουμε δομή χρόνου $\langle n, 1 \rangle$ για το $\pm\text{RMQ}$.

RMQ $\Rightarrow$ LCA

Αν $i^* = \operatorname{argmin}_{i \in [n]} A[i]$ τότε φτιάχνω ένα δέντρο με ρίζα το i^*

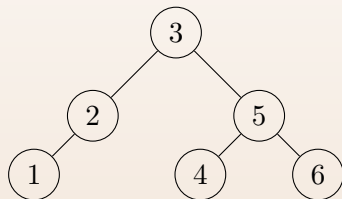
RMQ $\Rightarrow$ LCA

Αν $i^* = \operatorname{argmin}_{i \in [n]} A[i]$ τότε φτιάχνω ένα δέντρο με ρίζα το i^* στο αριστερό υπόδεντρο κάνω το ίδιο πράγμα αναδρομικά για τον πίνακα $A[1 \dots i^* - 1]$ και στο δεξί αναδρομικά για τον πίνακα $A[i^* + 1 \dots n]$.

RMQ $\Rightarrow$ LCA

Αν $i^* = \operatorname{argmin}_{i \in [n]} A[i]$ τότε φτιάχνω ένα δέντρο με ρίζα το i^* στο αριστερό υπόδεντρο κάνω το ίδιο πράγμα αναδρομικά για τον πίνακα $A[1 \dots i^* - 1]$ και στο δεξί αναδρομικά για τον πίνακα $A[i^* + 1 \dots n]$.

333	312	177	400	200	512
-----	-----	-----	-----	-----	-----

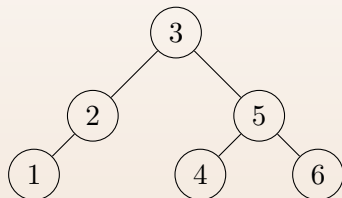


Έστω T το δέντρο που προκύπτει, το οποίο περιέχει δείκτες $i \in [n]$.

RMQ $\Rightarrow$ LCA

Αν $i^* = \operatorname{argmin}_{i \in [n]} A[i]$ τότε φτιάχνω ένα δέντρο με ρίζα το i^* στο αριστερό υπόδεντρο κάνω το ίδιο πράγμα αναδρομικά για τον πίνακα $A[1 \dots i^* - 1]$ και στο δεξί αναδρομικά για τον πίνακα $A[i^* + 1 \dots n]$.

333	312	177	400	200	512
-----	-----	-----	-----	-----	-----



Έστω T το δέντρο που προκύπτει, το οποίο περιέχει δείκτες $i \in [n]$.

Για κάθε ερώτηση (i, j) ο δείκτης που πιάνει το ελάχιστο στον πίνακα $A[i \dots j]$ είναι ο βαθύτερος κοινός πρόγονος των i, j στο T .

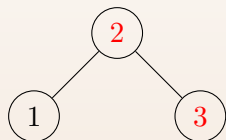
Πρέπει όμως να κατασκευάσω το T σε $O(n)$.

★ Αν έχω φτιάξει το σωστό δέντρο για τον πίνακα $A[1 \dots i]$ πως θα φτιάξω το σωστό δέντρο για τον πίνακα $A[1 \dots i + 1]$;

Πρέπει όμως να κατασκευάσω το T σε $O(n)$.

★ Αν έχω φτιάξει το σωστό δέντρο για τον πίνακα $A[1 \dots i]$ πως θα φτιάξω το σωστό δέντρο για τον πίνακα $A[1 \dots i + 1]$;

Στο παρακάτω παράδειγμα $A[3] < A[4] < A[2]$ και $A[5] < A[2]$.

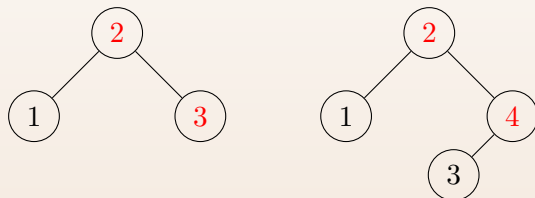


RMQ $\Rightarrow$ LCA

Πρέπει όμως να κατασκευάσω το T σε $O(n)$.

★ Αν έχω φτιάξει το σωστό δέντρο για τον πίνακα $A[1 \dots i]$ πως θα φτιάξω το σωστό δέντρο για τον πίνακα $A[1 \dots i + 1]$;

Στο παρακάτω παράδειγμα $A[3] < A[4] < A[2]$ και $A[5] < A[2]$.

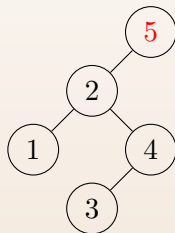
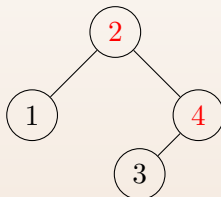
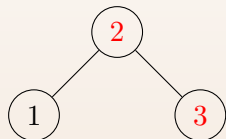


RMQ $\Rightarrow$ LCA

Πρέπει όμως να κατασκευάσω το T σε $O(n)$.

★ Αν έχω φτιάξει το σωστό δέντρο για τον πίνακα $A[1 \dots i]$ πως θα φτιάξω το σωστό δέντρο για τον πίνακα $A[1 \dots i + 1]$;

Στο παρακάτω παράδειγμα $A[3] < A[4] < A[2]$ και $A[5] < A[2]$.



👉 Οι κόκκινοι κόμβοι υποδεικνύουν ποιοι κόμβοι ενδεχομένως να χρειάζεται να μετακινηθούν με την εισαγωγή του επόμενου κόμβου.

RMQ $\Rightarrow$ LCA

Ιδέα: Κρατάω σε μια στοίβα S τους κόκκινους κόμβους,

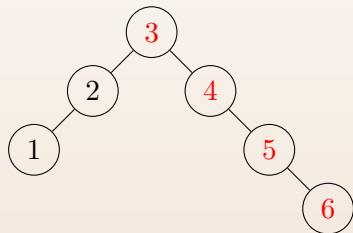
RMQ $\Rightarrow$ LCA

Ιδέα: Κρατάω σε μια στοίβα S τους **κόκκινους κόμβους**, δηλαδή το δεξιότερο μονοπάτι του τρέχοντος T

RMQ $\Rightarrow$ LCA

Ιδέα: Κρατάω σε μια στοίβα S τους **κόκκινους κόμβους**, δηλαδή το δεξιότερο μονοπάτι του τρέχοντος T

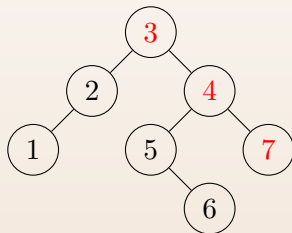
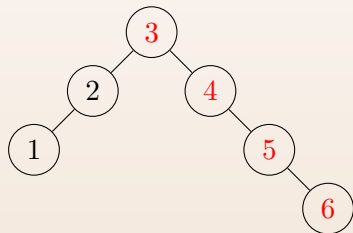
Έχουμε ότι $A[3] < A[4] < A[5] < A[6]$ και άρα ο 7 θα γίνει πατέρας ενός εκ των 3, 4, 5, 6 είτε δεξί παιδί του 6 (και δεν θα έχει κανένα δεξί παιδί).



RMQ $\Rightarrow$ LCA

Ιδέα: Κρατάω σε μια στοίβα S τους **κόκκινους κόμβους**, δηλαδή το δεξιότερο μονοπάτι του τρέχοντος T

Έχουμε ότι $A[3] < A[4] < A[5] < A[6]$ και άρα ο 7 θα γίνει πατέρας ενός εκ των 3, 4, 5, 6 είτε δεξί παιδί του 6 (και δεν θα έχει κανένα δεξί παιδί).



Υλοποίηση της γραμμικής κατασκευής του T .

- 1: Έστω μια στοίβα S και ένα κενό δέντρο T
- 2: Βάλε το 1 στην στοίβα και το 1 σαν ρίζα του T
- 3: Για $i = 2$ ως n
- 4: Όσο S μη κενή και $A[i] > A[S.top]$
- 5: Βγάλε το $S.top$ από το S
- 6: Βάλε το i στο T ώστε το
- 7: I. $S.top$ (αν υπάρχει) να γίνει αριστερό παιδί του i
- 8: II. το i να γίνει δεξί παιδί $S.top$ (αν υπάρχει)
- 9: Βάλε το i στην κορυφή της S

★ Κάθε πράξη του αλγόριθμου μπορεί να χρεωθεί σε δύο πράξεις: **είσοδος** δείκτη στην S ή **έξοδος** δείκτη από την S

Υλοποίηση της γραμμικής κατασκευής του T .

- 1: Έστω μια στοίβα S και ένα κενό δέντρο T
- 2: Βάλε το 1 στην στοίβα και το 1 σαν ρίζα του T
- 3: Για $i = 2$ ως n
- 4: Όσο S μη κενή και $A[i] > A[S.top]$
- 5: Βγάλε το $S.top$ από το S
- 6: Βάλε το i στο T ώστε το
- 7: I. $S.top$ (αν υπάρχει) να γίνει αριστερό παιδί του i
- 8: II. το i να γίνει δεξί παιδί $S.top$ (αν υπάρχει)
- 9: Βάλε το i στην κορυφή της S

★ Κάθε πράξη του αλγόριθμου μπορεί να χρεωθεί σε δύο πράξεις: **είσοδος** δείκτη στην S ή **έξοδος** δείκτη από την $S \Rightarrow O(n)$ χρόνος.

Υλοποίηση της γραμμικής κατασκευής του T .

- 1: Έστω μια στοίβα S και ένα κενό δέντρο T
- 2: Βάλε το 1 στην στοίβα και το 1 σαν ρίζα του T
- 3: Για $i = 2$ ως n
- 4: Όσο S μη κενή και $A[i] > A[S.top]$
- 5: Βγάλε το $S.top$ από το S
- 6: Βάλε το i στο T ώστε το
- 7: I. $S.top$ (αν υπάρχει) να γίνει αριστερό παιδί του i
- 8: II. το i να γίνει δεξί παιδί $S.top$ (αν υπάρχει)
- 9: Βάλε το i στην κορυφή της S

★ Κάθε πράξη του αλγόριθμου μπορεί να χρεωθεί σε δύο πράξεις: **είσοδος** δείκτη στην S ή **έξοδος** δείκτη από την $S \Rightarrow O(n)$ χρόνος.

👉 Υποψία αντισταθμιστικής ανάλυσης (επόμενη διάλεξη): Μπορεί για κάποιο i οι γραμμές 4 – 9 να είναι πολύ ακριβές,

Υλοποίηση της γραμμικής κατασκευής του T .

- 1: Έστω μια στοίβα S και ένα κενό δέντρο T
- 2: Βάλε το 1 στην στοίβα και το 1 σαν ρίζα του T
- 3: Για $i = 2$ ως n
- 4: Όσο S μη κενή και $A[i] > A[S.top]$
- 5: Βγάλε το $S.top$ από το S
- 6: Βάλε το i στο T ώστε το
- 7: I. $S.top$ (αν υπάρχει) να γίνει αριστερό παιδί του i
- 8: II. το i να γίνει δεξί παιδί $S.top$ (αν υπάρχει)
- 9: Βάλε το i στην κορυφή της S

★ Κάθε πράξη του αλγόριθμου μπορεί να χρεωθεί σε δύο πράξεις: **είσοδος** δείκτη στην S ή **έξοδος** δείκτη από την $S \Rightarrow O(n)$ χρόνος.

👉 Υποψία αντισταθμιστικής ανάλυσης (επόμενη διάλεξη): Μπορεί για κάποιο i οι γραμμές 4 – 9 να είναι πολύ ακριβές, αλλά τότε κάτι άλλο ‘καλό’ θα συμβεί

Υλοποίηση της γραμμικής κατασκευής του T .

- 1: Έστω μια στοίβα S και ένα κενό δέντρο T
- 2: Βάλε το 1 στην στοίβα και το 1 σαν ρίζα του T
- 3: Για $i = 2$ ως n
- 4: Όσο S μη κενή και $A[i] > A[S.top]$
- 5: Βγάλε το $S.top$ από το S
- 6: Βάλε το i στο T ώστε το
- 7: I. $S.top$ (αν υπάρχει) να γίνει αριστερό παιδί του i
- 8: II. το i να γίνει δεξί παιδί $S.top$ (αν υπάρχει)
- 9: Βάλε το i στην κορυφή της S

★ Κάθε πράξη του αλγόριθμου μπορεί να χρεωθεί σε δύο πράξεις: **είσοδος** δείκτη στην S ή **έξοδος** δείκτη από την $S \Rightarrow O(n)$ χρόνος.

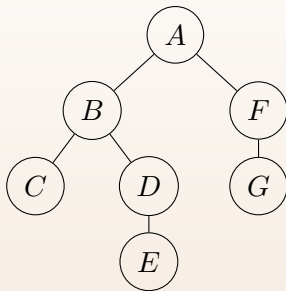
👉 Υποψία αντισταθμιστικής ανάλυσης (επόμενη διάλεξη): Μπορεί για κάποιο i οι γραμμές 4 – 9 να είναι πολύ ακριβές, αλλά τότε κάτι άλλο ‘καλό’ θα συμβεί $\Rightarrow$ θα μείνουν λίγοι κόκκινοι κόμβοι.

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος

$LCA \Rightarrow \pm RMQ$

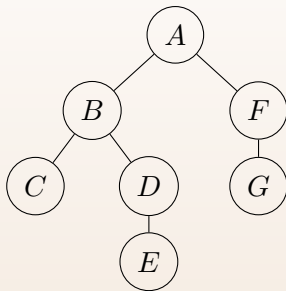
Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



A

$LCA \Rightarrow \pm RMQ$

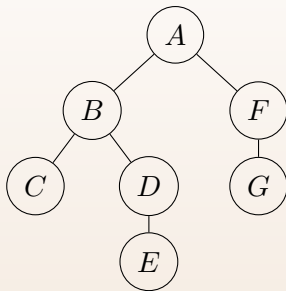
Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



A, B

$LCA \Rightarrow \pm RMQ$

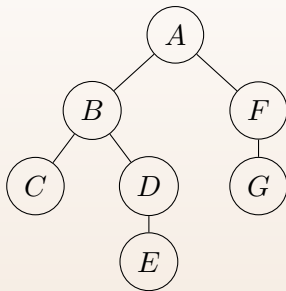
Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



A, B, C

$LCA \Rightarrow \pm RMQ$

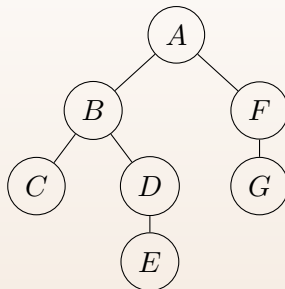
Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



A, B, C, B

$LCA \Rightarrow \pm RMQ$

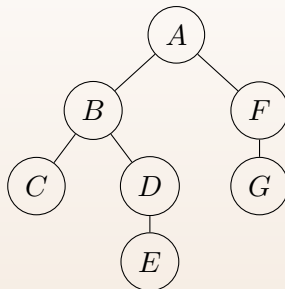
Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



A, B, C, B, D

$LCA \Rightarrow \pm RMQ$

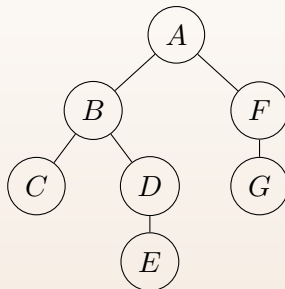
Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



A, B, C, B, D, E

$LCA \Rightarrow \pm RMQ$

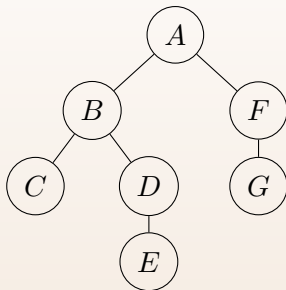
Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



$A, B, C, B, D, E, D, B, A, F, G, F, A$

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.

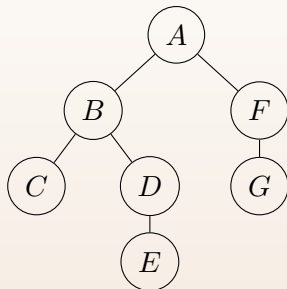


$A, B, C, B, D, E, D, B, A, F, G, F, A$

$L = [0$

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.

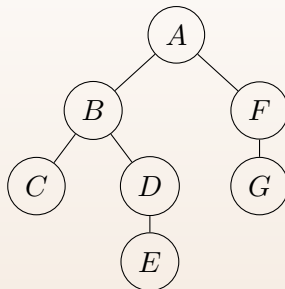


$A, B, C, B, D, E, D, B, A, F, G, F, A$

$L = [0, 1$

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.

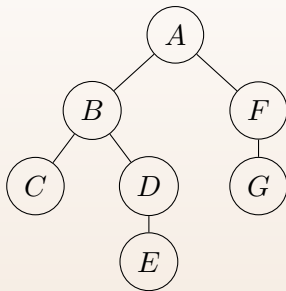


$A, B, C, B, D, E, D, B, A, F, G, F, A$

$L = [0, 1, 2$

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.

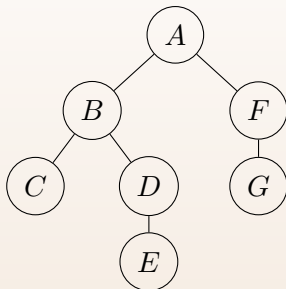


$A, B, C, B, D, E, D, B, A, F, G, F, A$

$L = [0, 1, 2, 1$

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.

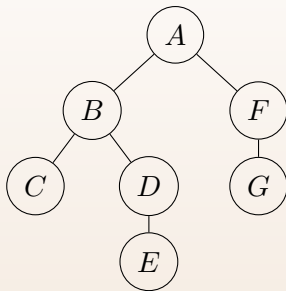


$A, B, C, B, D, E, D, B, A, F, G, F, A$

$L = [0, 1, 2, 1, 2$

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.

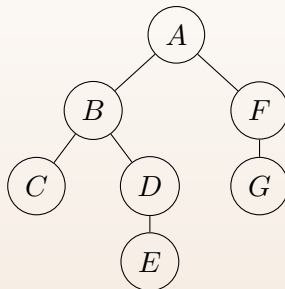


$A, B, C, B, D, E, D, B, A, F, G, F, A$

$L = [0, 1, 2, 1, 2, 3$

$LCA \Rightarrow \pm RMQ$

Διασχίζουμε το δέντρο T κατά βάθος και γράφουμε με επανάληψη κάτω τους κόμβους με τη σειρά που τους επισκεπτόμαστε, μαζί με το επίπεδο στο οποίο ανήκουν.



$A, B, C, B, D, E, D, B, A, F, G, F, A$

$L = [0, 1, 2, 1, 2, 3, 2, 1, 0, 1, 2, 1, 0]$

LCA $\Rightarrow$ $\pm$ RMQ

👁 Παρατήρηση: $|L[i + 1] - L[i]| = 1$,

LCA $\Rightarrow$ $\pm$ RMQ

👁 Παρατήρηση: $|L[i+1] - L[i]| = 1$, άρα μπορώ να χτίσω μία $\pm$ RMQ δομή πάνω στον L .

LCA $\Rightarrow$ $\pm$ RMQ

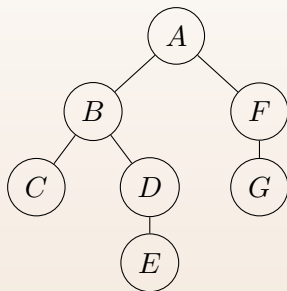
👁 Παρατήρηση: $|L[i + 1] - L[i]| = 1$, άρα μπορώ να χτίσω μία $\pm$ RMQ δομή πάνω στον L .

👉 Κρατάω επίσης μία θέση $I : T \rightarrow [2n - 1]$ κάθε κόμβου στον L .

LCA $\Rightarrow \pm\text{RMQ}$

👁 Παρατήρηση: $|L[i+1] - L[i]| = 1$, άρα μπορώ να χτίσω μία $\pm\text{RMQ}$ δομή πάνω στον L .

👉 Κρατάω επίσης μία θέση $I : T \rightarrow [2n - 1]$ κάθε κόμβου στον L .

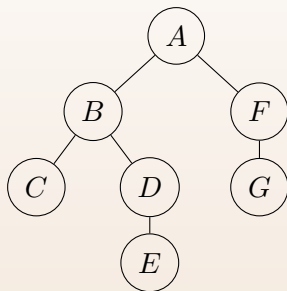


Για κάθε u, v κάνω
 $i := I[u], j := I[v]$

LCA $\Rightarrow \pm\text{RMQ}$

👁 Παρατήρηση: $|L[i+1] - L[i]| = 1$, άρα μπορώ να χτίσω μία $\pm\text{RMQ}$ δομή πάνω στον L .

👉 Κρατάω επίσης μία θέση $I : T \rightarrow [2n - 1]$ κάθε κόμβου στον L .

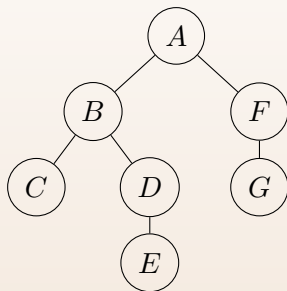


Για κάθε u, v κάνω
 $i := I[u], j := I[v]$ και ρωτάω
 $k^* := \operatorname{argmin}_{k \in [i, j]} L[k]$ στη $\pm\text{RMQ}$
δομή.

LCA $\Rightarrow \pm\text{RMQ}$

👁 Παρατήρηση: $|L[i+1] - L[i]| = 1$, άρα μπορώ να χτίσω μία $\pm\text{RMQ}$ δομή πάνω στον L .

👉 Κρατάω επίσης μία θέση $I : T \rightarrow [2n - 1]$ κάθε κόμβου στον L .

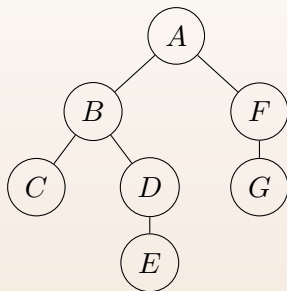


Για κάθε u, v κάνω
 $i := I[u], j := I[v]$ και ρωτάω
 $k^* := \operatorname{argmin}_{k \in [i, j]} L[k]$ στη $\pm\text{RMQ}$
δομή. Ο κόμβος στη θέση k^* του
 L , ήτοι ο $I^{-1}[k^*]$ είναι ο βαθύτερος
κοινός πρόγονος των u, v .

LCA $\Rightarrow \pm\text{RMQ}$

👁 Παρατήρηση: $|L[i+1] - L[i]| = 1$, άρα μπορώ να χτίσω μία $\pm\text{RMQ}$ δομή πάνω στον L .

👉 Κρατάω επίσης μία θέση $I : T \rightarrow [2n - 1]$ κάθε κόμβου στον L .



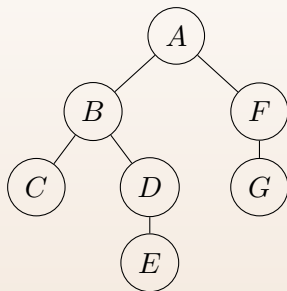
Για κάθε u, v κάνω
 $i := I[u], j := I[v]$ και ρωτάω
 $k^* := \operatorname{argmin}_{k \in [i, j]} L[k]$ στη $\pm\text{RMQ}$
δομή. Ο κόμβος στη θέση k^* του
 L , ήτοι ο $I^{-1}[k^*]$ είναι ο βαθύτερος
κοινός πρόγονος των u, v .

$A, B, \textcolor{red}{C}, \underline{B}, D, \textcolor{red}{E}, D, B, A, F, G, F, A$
 $L = [0, 1, \textcolor{red}{2}, \underline{1}, 2, \textcolor{red}{3}, 2, 1, 0, 1, 2, 1, 0]$

LCA $\Rightarrow \pm\text{RMQ}$

👁 Παρατήρηση: $|L[i+1] - L[i]| = 1$, άρα μπορώ να χτίσω μία $\pm\text{RMQ}$ δομή πάνω στον L .

👉 Κρατάω επίσης μία θέση $I : T \rightarrow [2n-1]$ κάθε κόμβου στον L .



Για κάθε u, v κάνω
 $i := I[u], j := I[v]$ και ρωτάω
 $k^* := \operatorname{argmin}_{k \in [i, j]} L[k]$ στη $\pm\text{RMQ}$
δομή. Ο κόμβος στη θέση k^* του
 L , ήτοι ο $I^{-1}[k^*]$ είναι ο βαθύτερος
κοινός πρόγονος των u, v .

$A, B, \textcolor{red}{C}, \underline{B}, D, \textcolor{red}{E}, D, B, A, F, G, F, A$
 $L = [0, 1, \textcolor{red}{2}, \underline{1}, 2, \textcolor{red}{3}, 2, 1, 0, 1, 2, 1, 0]$

Υπάρχει μια λεπτομέρεια εδώ, αλλά ο Διάβολος κρύβεται πάντα στις λεπτομέρειες.

Λύνοντας το $\pm$ RMQ

1	2	3	4	3	2	1	2	3	2
---	---	---	---	---	---	---	---	---	---

Μπορώ να ορίσω τον μήκους $n - 1$ πίνακα Δ με $\Delta[i] = A[i + 1] - A[i]$.

Λύνοντας το $\pm\text{RMQ}$

1	2	3	4	3	2	1	2	3	2
---	---	---	---	---	---	---	---	---	---

Μπορώ να ορίσω τον μήκους $n - 1$ πίνακα Δ με $\Delta[i] = A[i + 1] - A[i]$.

1	1	1	-1	-1	-1	1	1	-1
---	---	---	----	----	----	---	---	----

Δύο πίνακες A, A' με ίδιο πίνακα διαφορών θα λέγονται ισοδύναμοι.

Λύνοντας το $\pm\text{RMQ}$

1	2	3	4	3	2	1	2	3	2
---	---	---	---	---	---	---	---	---	---

Μπορώ να ορίσω τον μήκους $n - 1$ πίνακα Δ με $\Delta[i] = A[i + 1] - A[i]$.

1	1	1	-1	-1	-1	1	1	-1
---	---	---	----	----	----	---	---	----

Δύο πίνακες A, A' με ίδιο πίνακα διαφορών θα λέγονται ισοδύναμοι.

❶ $k^* = \operatorname{argmin}_{k \in [i, j]} A[i, j] = \operatorname{argmin}_{k \in [i, j]} A'[i, j]$

Λύνοντας το $\pm\text{RMQ}$

1	2	3	4	3	2	1	2	3	2
---	---	---	---	---	---	---	---	---	---

Μπορώ να ορίσω τον μήκους $n - 1$ πίνακα Δ με $\Delta[i] = A[i + 1] - A[i]$.

1	1	1	-1	-1	-1	1	1	-1
---	---	---	----	----	----	---	---	----

Δύο πίνακες A, A' με ίδιο πίνακα διαφορών θα λέγονται ισοδύναμοι.

- 1 $k^* = \operatorname{argmin}_{k \in [i, j]} A[i, j] = \operatorname{argmin}_{k \in [i, j]} A'[i, j]$
- 2 $A[k^*] = A'[k^*] + (A'[1] - A[1])$.

Λύνοντας το $\pm\text{RMQ}$

1	2	3	4	3	2	1	2	3	2
---	---	---	---	---	---	---	---	---	---

Μπορώ να ορίσω τον μήκους $n - 1$ πίνακα Δ με $\Delta[i] = A[i + 1] - A[i]$.

1	1	1	-1	-1	-1	1	1	-1
---	---	---	----	----	----	---	---	----

Δύο πίνακες A, A' με ίδιο πίνακα διαφορών θα λέγονται ισοδύναμοι.

- 1 $k^* = \operatorname{argmin}_{k \in [i, j]} A[i, j] = \operatorname{argmin}_{k \in [i, j]} A'[i, j]$
- 2 $A[k^*] = A'[k^*] + (A'[1] - A[1])$.

Κοινώς, αν απαντήσω μία ερώτηση στον A την έχω απαντήσει και στον A' .

Εκμετάλλευση της ισοδυναμίας.

👁 Για κάθε b υπάρχουν επί της ουσίας μόνο 2^{b-1} πίνακες μήκους b για το $\pm\text{RMQ}$ πρόβλημα.

Εκμετάλλευση της ισοδυναμίας.

👁 Για κάθε b υπάρχουν επί της ουσίας μόνο 2^{b-1} πίνακες μήκους b για το $\pm\text{RMQ}$ πρόβλημα.

Άρα ίσως να μπορώ να προϋπολογίσω την απάντηση για όλους τους μικρούς πίνακες.

Εκμετάλλευση της ισοδυναμίας.

👁 Για κάθε b υπάρχουν επί της ουσίας μόνο 2^{b-1} πίνακες μήκους b για το $\pm\text{RMQ}$ πρόβλημα.

Άρα ίσως να μπορώ να προϋπολογίσω την απάντηση για όλους τους μικρούς πίνακες. Σπάω τον πίνακα A σε $\frac{n}{b}$ μπλοκ μεγέθους b και παίρνω τον A_{blocked} με $A_{\text{blocked}}[i] = \min_{k \in [(i-1) \cdot b + 1, i \cdot b]} A[k]$.

Εκμετάλλευση της ισοδυναμίας.

👁 Για κάθε b υπάρχουν επί της ουσίας μόνο 2^{b-1} πίνακες μήκους b για το $\pm\text{RMQ}$ πρόβλημα.

Άρα ίσως να μπορώ να προϋπολογίσω την απάντηση για όλους τους μικρούς πίνακες. Σπάω τον πίνακα A σε $\frac{n}{b}$ μπλοκ μεγέθους b και παίρνω τον A_{blocked} με $A_{\text{blocked}}[i] = \min_{k \in [(i-1) \cdot b + 1, i \cdot b]} A[k]$.

Φτιάχνω τη δομή RMQ της προηγούμενης διάλεξης στον A_{blocked}

Εκμετάλλευση της ισοδυναμίας.

👁 Για κάθε b υπάρχουν επί της ουσίας μόνο 2^{b-1} πίνακες μήκους b για το $\pm\text{RMQ}$ πρόβλημα.

Άρα ίσως να μπορώ να προϋπολογίσω την απάντηση για όλους τους μικρούς πίνακες. Σπάω τον πίνακα A σε $\frac{n}{b}$ μπλοκ μεγέθους b και παίρνω τον A_{blocked} με $A_{\text{blocked}}[i] = \min_{k \in [(i-1) \cdot b + 1, i \cdot b]} A[k]$.

Φτιάχνω τη δομή RMQ της προηγούμενης διάλεξης στον A_{blocked} με χρόνο $\langle \frac{n}{b} \cdot \log \frac{n}{b}, 1 \rangle$.

1	2	3	4	3	2	1	2	3	2	3	4
---	---	---	---	---	---	---	---	---	---	---	---

Πως θα απαντηθεί μία ερώτηση.

1	2	3	4		3	2	1	2		3	2	3	4
---	---	---	---	--	---	---	---	---	--	---	---	---	---

Μία ερώτηση $\text{RMQ}(i, j)$ (πχ εδώ η $\text{RMQ}(3, 10)$) σπάει σε τρία διαστήματα.

Πως θα απαντηθεί μία ερώτηση.

1	2	3	4		3	2	1	2		3	2	3	4
---	---	---	---	--	---	---	---	---	--	---	---	---	---

Μία ερώτηση $\text{RMQ}(i, j)$ (πχ εδώ η $\text{RMQ}(3, 10)$) σπάει σε τρία διαστήματα.

- 1 Τα πράσινο κομμάτι μπορεί να προσομοιωθεί με μία ερώτηση στον A_{blocked} σε $O(1)$ χρόνο. ✓

Πως θα απαντηθεί μία ερώτηση.

1	2	3	4		3	2	1	2		3	2	3	4
---	---	---	---	--	---	---	---	---	--	---	---	---	---

Μία ερώτηση $\text{RMQ}(i, j)$ (πχ εδώ η $\text{RMQ}(3, 10)$) σπάει σε τρία διαστήματα.

- 1 Τα πράσινο κομμάτι μπορεί να προσομοιωθεί με μία ερώτηση στον A_{blocked} σε $O(1)$ χρόνο. ✓
- 2 Το μπλε κομμάτι το οποίο είναι ένα διάστημα το οποίο περιέχεται σε ένα από τα μικρά μπλοκ μεγέθους b .

Πως θα απαντηθεί μία ερώτηση.

1	2	3	4	3	2	1	2	3	2	3	4
---	---	---	---	---	---	---	---	---	---	---	---

Μία ερώτηση $\text{RMQ}(i, j)$ (πχ εδώ η $\text{RMQ}(3, 10)$) σπάει σε τρία διαστήματα.

- 1 Τα πράσινο κομμάτι μπορεί να προσομοιωθεί με μία ερώτηση στον A_{blocked} σε $O(1)$ χρόνο. ✓
- 2 Το μπλε κομμάτι το οποίο είναι ένα διάστημα το οποίο περιέχεται σε ένα από τα μικρά μπλοκ μεγέθους b .
- 3 Το κόκκινο κομμάτι για το οποίο ισχύει το ίδιο.

Πως θα απαντηθεί μία ερώτηση.

1	2	3	4	3	2	1	2	3	2	3	4
---	---	---	---	---	---	---	---	---	---	---	---

Μία ερώτηση $\text{RMQ}(i, j)$ (πχ εδώ η $\text{RMQ}(3, 10)$) σπάει σε τρία διαστήματα.

- 1 Τα πράσινο κομμάτι μπορεί να προσομοιωθεί με μία ερώτηση στον A_{blocked} σε $O(1)$ χρόνο. ✓
- 2 Το μπλε κομμάτι το οποίο είναι ένα διάστημα το οποίο περιέχεται σε ένα από τα μικρά μπλοκ μεγέθους b .
- 3 Το κόκκινο κομμάτι για το οποίο ισχύει το ίδιο.

Δεν μπορούμε απλά να δοκιμάσουμε όλο το διάστημα μέσα στο μπλοκ γιατί θα πάρει $O(b)$ χρόνο.

Πως θα απαντηθεί μία ερώτηση.

1	2	3	4	3	2	1	2	3	2	3	4
---	---	---	---	---	---	---	---	---	---	---	---

Μία ερώτηση $\text{RMQ}(i, j)$ (πχ εδώ η $\text{RMQ}(3, 10)$) σπάει σε τρία διαστήματα.

- 1 Τα πράσινο κομμάτι μπορεί να προσομοιωθεί με μία ερώτηση στον A_{blocked} σε $O(1)$ χρόνο. ✓
- 2 Το μπλε κομμάτι το οποίο είναι ένα διάστημα το οποίο περιέχεται σε ένα από τα μικρά μπλοκ μεγέθους b .
- 3 Το κόκκινο κομμάτι για το οποίο ισχύει το ίδιο.

Δεν μπορούμε απλά να δοκιμάσουμε όλο το διάστημα μέσα στο μπλοκ γιατί θα πάρει $O(b)$ χρόνο. Αν $b = O(1)$ τότε ο χρόνος προεπεξεργασίας γίνεται $\Theta(n \log n)$,

Πως θα απαντηθεί μία ερώτηση.

1	2	3	4	3	2	1	2	3	2	3	4
---	---	---	---	---	---	---	---	---	---	---	---

Μία ερώτηση $\text{RMQ}(i, j)$ (πχ εδώ η $\text{RMQ}(3, 10)$) σπάει σε τρία διαστήματα.

- 1 Τα πράσινο κομμάτι μπορεί να προσομοιωθεί με μία ερώτηση στον A_{blocked} σε $O(1)$ χρόνο. ✓
- 2 Το μπλε κομμάτι το οποίο είναι ένα διάστημα το οποίο περιέχεται σε ένα από τα μικρά μπλοκ μεγέθους b .
- 3 Το κόκκινο κομμάτι για το οποίο ισχύει το ίδιο.

Δεν μπορούμε απλά να δοκιμάσουμε όλο το διάστημα μέσα στο μπλοκ γιατί θα πάρει $O(b)$ χρόνο. Αν $b = O(1)$ τότε ο χρόνος προεπεξεργασίας γίνεται $\Theta(n \log n)$, αλλιώς έχουμε $\omega(1)$ χρόνο ερώτησης.

Εκμετάλλευση της ισοδυναμίας.

👁 Υπάρχουν επί της ουσίας 2^{b-1} διαφορετικοί υπο-πίνακες που μας ενδιαφέρουν.

Εκμετάλλευση της ισοδυναμίας.

👁 Υπάρχουν επί της ουσίας 2^{b-1} διαφορετικοί υπο-πίνακες που μας ενδιαφέρουν. Άρα αν βάζαμε $b = \lceil \frac{1}{2} \log n \rceil$ θα είχαμε επί της ουσίας ο πολύ $\sqrt{n}$ διαφορετικά μπλοκ:

Εκμετάλλευση της ισοδυναμίας.

👁 Υπάρχουν επί της ουσίας 2^{b-1} διαφορετικοί υπο-πίνακες που μας ενδιαφέρουν. Άρα αν βάζαμε $b = \lceil \frac{1}{2} \log n \rceil$ θα είχαμε επί της ουσίας ο πολύ $\sqrt{n}$ διαφορετικά μπλοκ: από τα $\frac{n}{b} \approx \frac{n}{\log n}$ μπλοκ του A η συντριπτική πλειοψηφία επαναλαμβάνεται πολύ συχνά!

Εκμετάλλευση της ισοδυναμίας.

👁 Υπάρχουν επί της ουσίας 2^{b-1} διαφορετικοί υπο-πίνακες που μας ενδιαφέρουν. Άρα αν βάζαμε $b = \lceil \frac{1}{2} \log n \rceil$ θα είχαμε επί της ουσίας ο πολύ $\sqrt{n}$ διαφορετικά μπλοκ: από τα $\frac{n}{b} \approx \frac{n}{\log n}$ μπλοκ του A η συντριπτική πλειοψηφία επαναλαμβάνεται πολύ συχνά!

Εκμετάλλευση: Για καθεμία από τις 2^{b-1} κλάσεις ισοδυναμίας Δ των πινάκων, διαλέγουμε έναν αντιπρόσωπο, και για κάθε πιθανή ερώτηση $(i, j) \in [b] \times [b]$ **προ-υπολογίζουμε** την απάντηση και την αποθηκεύουμε στο $P[\Delta][i][j]$.

Εκμετάλλευση της ισοδυναμίας.

👁 Υπάρχουν επί της ουσίας 2^{b-1} διαφορετικοί υπο-πίνακες που μας ενδιαφέρουν. Άρα αν βάζαμε $b = \lceil \frac{1}{2} \log n \rceil$ θα είχαμε επί της ουσίας ο πολύ $\sqrt{n}$ διαφορετικά μπλοκ: από τα $\frac{n}{b} \approx \frac{n}{\log n}$ μπλοκ του A η συντριπτική πλειοψηφία επαναλαμβάνεται πολύ συχνά!

Εκμετάλλευση: Για καθεμία από τις 2^{b-1} κλάσεις ισοδυναμίας Δ των πινάκων, διαλέγουμε έναν αντιπρόσωπο, και για κάθε πιθανή ερώτηση $(i, j) \in [b] \times [b]$ **προ-υπολογίζουμε** την απάντηση και την αποθηκεύουμε στο $P[\Delta][i][j]$. Χρόνος $O(2^b \cdot b^3)$.

Εκμετάλλευση της ισοδυναμίας.

👁 Υπάρχουν επί της ουσίας 2^{b-1} διαφορετικοί υπο-πίνακες που μας ενδιαφέρουν. Άρα αν βάζαμε $b = \lceil \frac{1}{2} \log n \rceil$ θα είχαμε επί της ουσίας ο πολύ $\sqrt{n}$ διαφορετικά μπλοκ: από τα $\frac{n}{b} \approx \frac{n}{\log n}$ μπλοκ του A η συντριπτική πλειοψηφία επαναλαμβάνεται πολύ συχνά!

Εκμετάλλευση: Για καθεμία από τις 2^{b-1} κλάσεις ισοδυναμίας Δ των πινάκων, διαλέγουμε έναν αντιπρόσωπο, και για κάθε πιθανή ερώτηση $(i, j) \in [b] \times [b]$ **προ-υπολογίζουμε** την απάντηση και την αποθηκεύουμε στο $P[\Delta][i][j]$. Χρόνος $O(2^b \cdot b^3)$.

Οπότε, όποτε πρέπει να κάνουμε ερώτηση σε έναν μπλοκ μεγέθους b , αρκεί να ξέρουμε το Δ του και να κοιτάζουμε τη σωστή εγγραφή του πίνακα P !

1	2	3	4	3	2	1	2	3	2	3	4
---	---	---	---	---	---	---	---	---	---	---	---

Λεπτομέρειες υλοποίησης και χρόνος εκτέλεσης.

Ας διαλέξουμε $b = \lceil \frac{1}{2} \log n \rceil$

- 1 Για κάθε μπλοκ προυπολογίζω το Δ του και το κρατάω σαν bitstring (σημαντικό για $O(1)$ χρόνο!).
- 2 Για κάθε $i \in [n]$ κρατάω δείκτες στο τέλος και στην αρχή του μπλοκ το οποίο βρίσκεται.

Λεπτομέρειες υλοποίησης και χρόνος εκτέλεσης.

Ας διαλέξουμε $b = \lceil \frac{1}{2} \log n \rceil$

- 1 Για κάθε μπλοκ προυπολογίζω το Δ του και το κρατάω σαν bitstring (σημαντικό για $O(1)$ χρόνο!).
- 2 Για κάθε $i \in [n]$ κρατάω δείκτες στο τέλος και στην αρχή του μπλοκ το οποίο βρίσκεται.
- 3 Έτσι μπορώ να σπάσω κάθε διάστημα $[i, j]$ σε το πολύ τρία κομμάτια σε $O(1)$ χρόνο και να κάνω τις αντίστοιχες ερωτήσεις.

Λεπτομέρειες υλοποίησης και χρόνος εκτέλεσης.

Ας διαλέξουμε $b = \lceil \frac{1}{2} \log n \rceil$

- 1 Για κάθε μπλοκ προυπολογίζω το Δ του και το κρατάω σαν bitstring (σημαντικό για $O(1)$ χρόνο!).
- 2 Για κάθε $i \in [n]$ κρατάω δείκτες στο τέλος και στην αρχή του μπλοκ το οποίο βρίσκεται.
- 3 Έτσι μπορώ να σπάσω κάθε διάστημα $[i, j]$ σε το πολύ τρία κομμάτια σε $O(1)$ χρόνο και να κάνω τις αντίστοιχες ερωτήσεις.

Ο χρόνος προϋπολογισμού είναι

$$O\left(\frac{n}{b} \cdot \log\left(\frac{n}{b}\right) + 2^b \cdot b^3\right)$$

Λεπτομέρειες υλοποίησης και χρόνος εκτέλεσης.

Ας διαλέξουμε $b = \lceil \frac{1}{2} \log n \rceil$

- 1 Για κάθε μπλοκ προυπολογίζω το Δ του και το κρατάω σαν bitstring (σημαντικό για $O(1)$ χρόνο!).
- 2 Για κάθε $i \in [n]$ κρατάω δείκτες στο τέλος και στην αρχή του μπλοκ το οποίο βρίσκεται.
- 3 Έτσι μπορώ να σπάσω κάθε διάστημα $[i, j]$ σε το πολύ τρία κομμάτια σε $O(1)$ χρόνο και να κάνω τις αντίστοιχες ερωτήσεις.

Ο χρόνος προϋπολογισμού είναι

$$O\left(\frac{n}{b} \cdot \log\left(\frac{n}{b}\right) + 2^b \cdot b^3\right) = O(n).$$

Λεπτομέρειες υλοποίησης και χρόνος εκτέλεσης.

Ας διαλέξουμε $b = \lceil \frac{1}{2} \log n \rceil$

- 1 Για κάθε μπλοκ προυπολογίζω το Δ του και το κρατάω σαν bitstring (σημαντικό για $O(1)$ χρόνο!).
- 2 Για κάθε $i \in [n]$ κρατάω δείκτες στο τέλος και στην αρχή του μπλοκ το οποίο βρίσκεται.
- 3 Έτσι μπορώ να σπάσω κάθε διάστημα $[i, j]$ σε το πολύ τρία κομμάτια σε $O(1)$ χρόνο και να κάνω τις αντίστοιχες ερωτήσεις.

Ο χρόνος προϋπολογισμού είναι

$$O\left(\frac{n}{b} \cdot \log\left(\frac{n}{b}\right) + 2^b \cdot b^3\right) = O(n).$$

Αυτο ολοκληρώνει τον αλγόριθμο και δείχνει ότι και τα δύο προβλήματα LCA, RMQ επιδέχονται δομές δεδομένων χρόνου $\langle n, 1 \rangle$.