Προηγμένα Θέματα Αλγορίθμων

Διαλέξη 1: Δομές Δεδομένων Ι

Βασίλειος Νάκος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Αθηνών

Δομές Δεδομένων RMQ και LCA.

Θα δούμε δύο **στατικά** προβλήματα δομών δεδομένων: το πρόβλημα RMQ και το πρόβλημα LCA.

Δομές Δεδομένων RMQ και LCA.

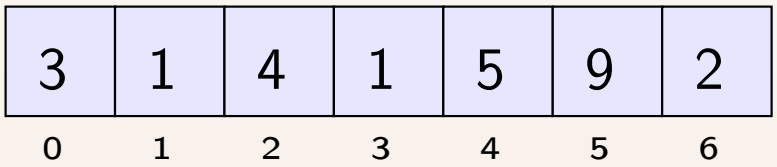
Θα δούμε δύο **στατικά** προβλήματα δομών δεδομένων: το πρόβλημα RMQ και το πρόβλημα LCA.

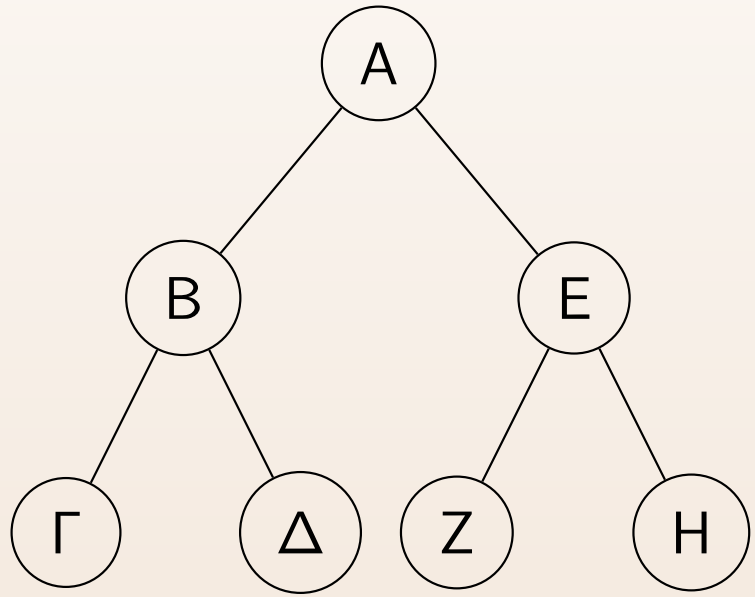
$$\text{RMQ} \Rightarrow \text{πίνακες, LCA} \Rightarrow \text{δέντρα.}$$

Δομές Δεδομένων RMQ και LCA.

Θα δούμε δύο **στατικά** προβλήματα δομών δεδομένων: το πρόβλημα RMQ και το πρόβλημα LCA.

RMQ ⇒πίνακες, LCA ⇒ δέντρα.



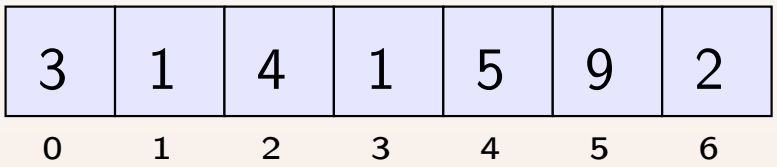


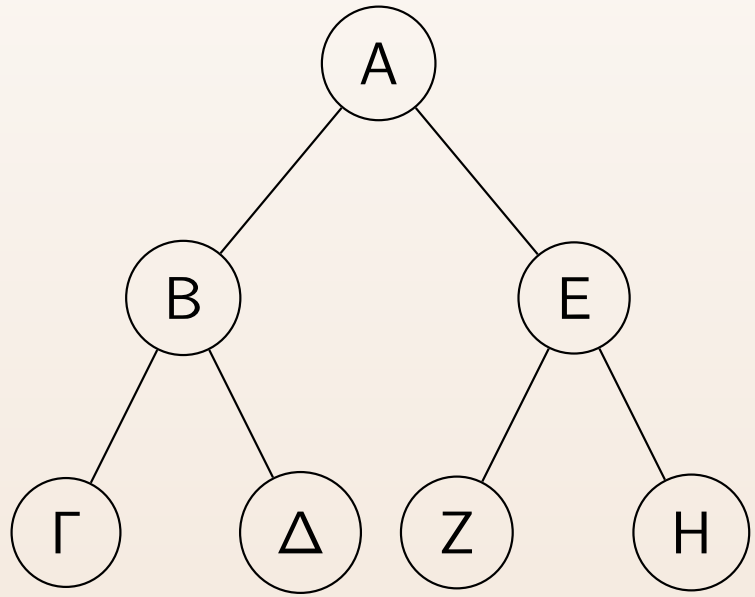
Θα λέμε ότι μια δομή έχει χρόνο $\langle f, g \rangle$ αν μπορεί να χτιστεί σε χρόνο $O(f)$ και να απαντάει ερωτήσεις σε χρόνο $O(g)$.

Δομές Δεδομένων RMQ και LCA.

Θα δούμε δύο **στατικά** προβλήματα δομών δεδομένων: το πρόβλημα RMQ και το πρόβλημα LCA.

RMQ ⇒πίνακες, LCA ⇒ δέντρα.





Θα λέμε ότι μια δομή έχει χρόνο $\langle f, g \rangle$ αν μπορεί να χτιστεί σε χρόνο $O(f)$ και να απαντάει ερωτήσεις σε χρόνο $O(g)$.

Πρόβλημα RMQ

Μας δίνεται στατικός πίνακας A μήκους n και θέλουμε να απαντάμε ερωτήματα της μορφής

$$\mathrm{RMQ}(i,j) = \min_{k \in [i,j]} A[k].$$

Πρόβλημα RMQ

Μας δίνεται στατικός πίνακας A μήκους n και θέλουμε να απαντάμε ερωτήματα της μορφής

$$\mathrm{RMQ}(i,j) = \min_{k\in[i,j]} A[k].$$

RMQ(2, 5)

4	7	1	3	9	2	6	5
0	1	2	3	4	5	6	7

RMQ(0,6)

4	7	1	3	9	2	6	5
0	1	2	3	4	5	6	7

Πρόβλημα LCA

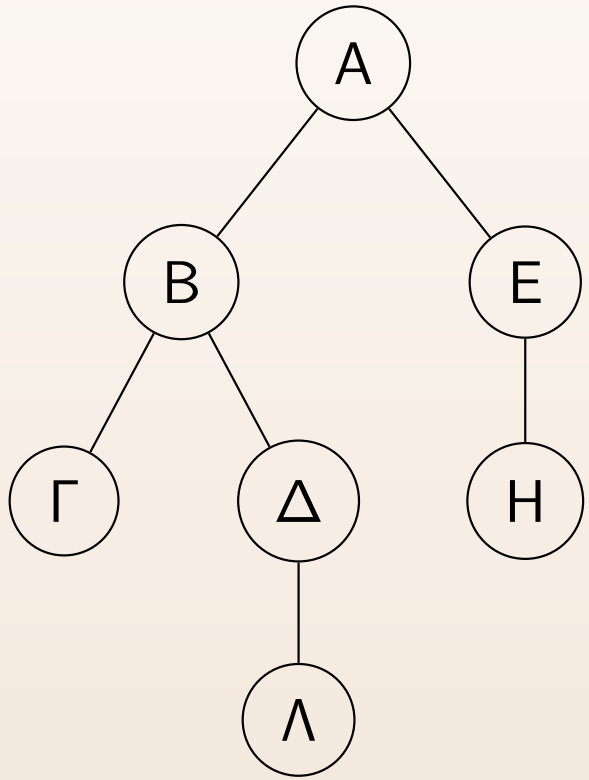
Μας δίνεται στατικό δέντρο T μεγέθους n με δεδομένη ρίζα και θέλουμε να απαντήσουμε ερωτήματα της μορφής $\mathrm{LCA}(u, v) =$ ποιος είναι ο κοινός πρόγονος μέγιστου βάθους των u, v ;

Πρόβλημα LCA

Μας δίνεται στατικό δέντρο T μεγέθους n με δεδομένη ρίζα και θέλουμε να απαντήσουμε ερωτήματα της μορφής $\mathrm{LCA}(u, v) =$ ποιος είναι ο κοινός πρόγονος μέγιστου βάθους των u, v ;

$\mathrm{LCA}(\Gamma,\Lambda) =$B $\quad \mathrm{LCA}(\Lambda,H) =$A
$\mathrm{LCA}(\Lambda,$B$) =$B $\quad \mathrm{LCA}($A,B$) =$A

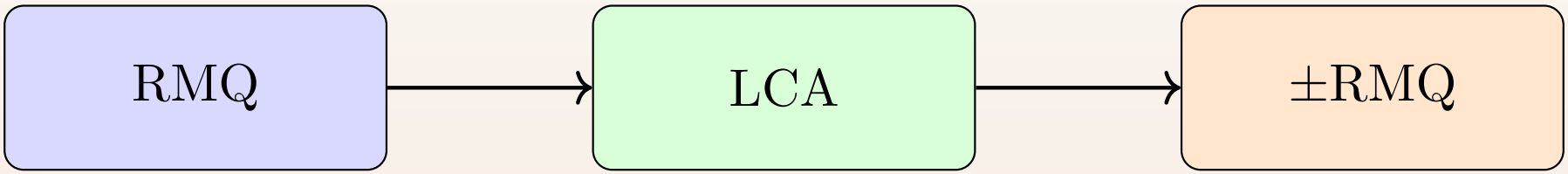
Το δέντρο **δεν** αλλάζει, και έχουμε χρόνο $O(f)$ να το επεξεργαστούμε και χρόνο $O(g)$ να απαντήσουμε κάθε ερώτηση. Πόσο μικρά μπορούν να γίνουν τα $f := f(n), g := g(n)$;



Σχέση μεταξύ RMQ και LCA.

$\pm$RMQ : Το ίδιο πρόβλημα με το RMQ αλλά για τον πίνακα A ισχύει επιπρόσθετα ότι $|A[i+1] - A[i]| = 1$.

Θα δείξουμε δύο 'γρήγορες' αναγωγές:



Στο τέλος των δύο διαλέξεων θα δείξουμε μία *αυτο-αναγωγή* από το RMQ στον εαυτό του, για την ακρίβεια στην ειδική περίπτωση του $\pm$RMQ χρησιμοποιώντας ως ενδιάμεσο σταθμό το LCA.

$\star$ Και λύνοντας το ευκολότερο $\pm$RMQ θα κατασκευάσουμε δομή δεδομένων με χρόνο $\langle n, 1 \rangle$ και για τα δύο αρχικά προβλήματα.

Μία πρώτη λύση για το RMQ.

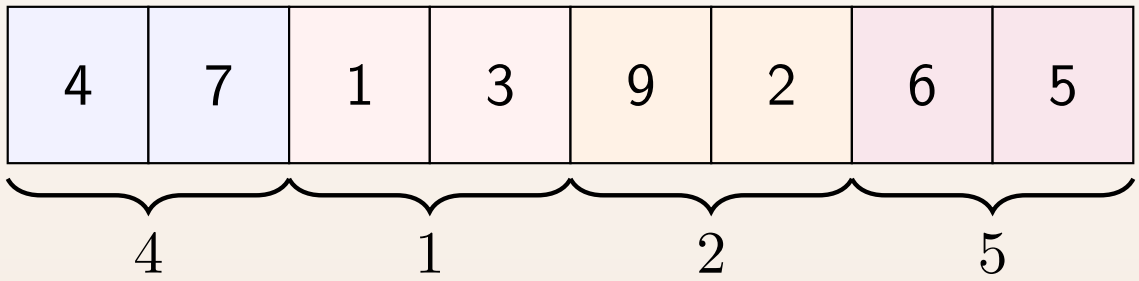
Έστω μία παράμετρος $B|n$ και υπολογίζουμε για κάθε $j \in \left[\frac{n}{B}\right]$ το

$$\mathrm{A}_{\mathrm{blocked}}[j] = \min_{(j-1)B+1 \leq i \leq jB} A[i].$$

Μία πρώτη λύση για το RMQ.

Έστω μία παράμετρος $B|n$ και υπολογίζουμε για κάθε $j \in \left[\frac{n}{B}\right]$ το

$$\mathrm{A}_{\mathrm{blocked}}[j] = \min_{(j-1)B+1 \le i \le jB} A[i].$$

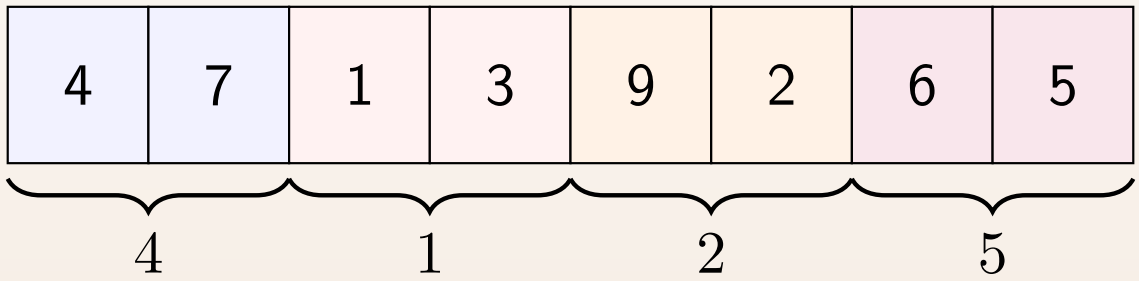


Στην προκειμένη περίπτωση για $B = 2$ έχουμε $\mathrm{A}_{\mathrm{blocked}} = [4, 1, 2, 5]$.

Μία πρώτη λύση για το RMQ.

Έστω μία παράμετρος $B|n$ και υπολογίζουμε για κάθε $j \in \left[\frac{n}{B}\right]$ το

$$\mathrm{A}_{\mathrm{blocked}}[j] = \min_{(j-1)B+1 \le i \le jB} A[i].$$

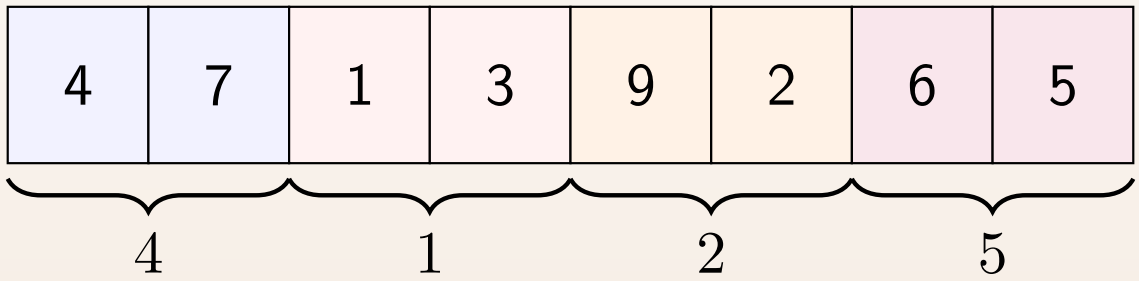


Στην προκειμένη περίπτωση για $B = 2$ έχουμε $\mathrm{A}_{\mathrm{blocked}} = [4, 1, 2, 5]$.
Αν είχαμε $B = 4$ θα είχαμε $\mathrm{A}_{\mathrm{blocked}} = [1, 2]$.

Μία πρώτη λύση για το RMQ.

Έστω μία παράμετρος $B|n$ και υπολογίζουμε για κάθε $j \in \left[\frac{n}{B}\right]$ το

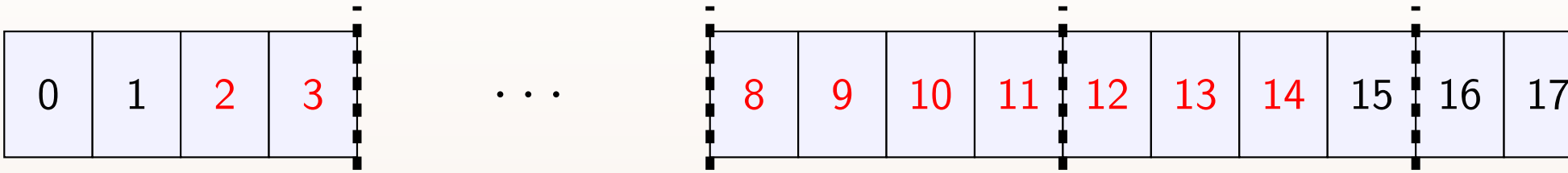
$$\mathrm{A}_{\mathrm{blocked}}[j] = \min_{(j-1)B+1 \le i \le jB} A[i].$$



Στην προκειμένη περίπτωση για $B = 2$ έχουμε $\mathrm{A}_{\mathrm{blocked}} = [4, 1, 2, 5]$.
Αν είχαμε $B = 4$ θα είχαμε $\mathrm{A}_{\mathrm{blocked}} = [1, 2]$.

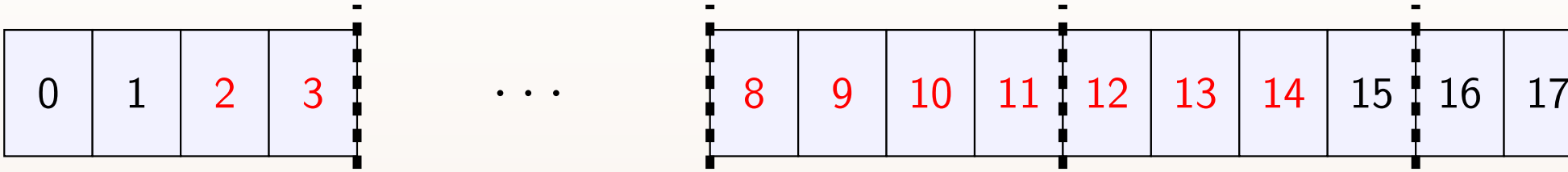
Πως θα απαντούσαμε μία ερώτηση $\mathrm{RMQ}(i, j)$ έχοντας τον $\mathrm{A}_{\mathrm{blocked}}$;

Με τη βοήθεια του A_{blocked}



Για κάθε ερώτηση $\text{RMQ}(i, j)$

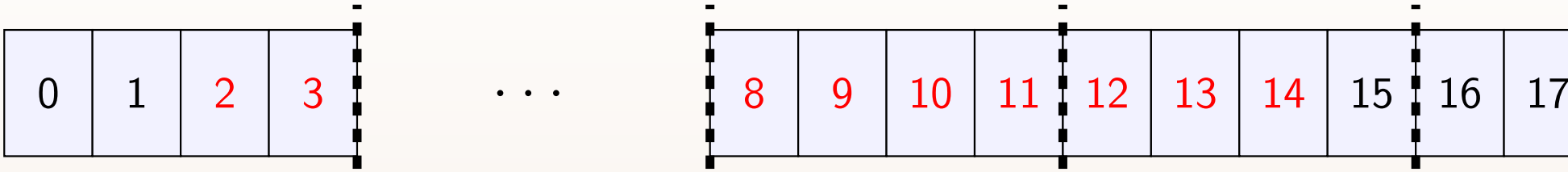
Με τη βοήθεια του A_{blocked}



Για κάθε ερώτηση $\text{RMQ}(i, j)$ γράφω $[i, j] = I_1 \cup I_2 \cup I_3$ ώστε

1. Το I_2 είναι της μορφής $[i' \cdot B, j' \cdot B]$

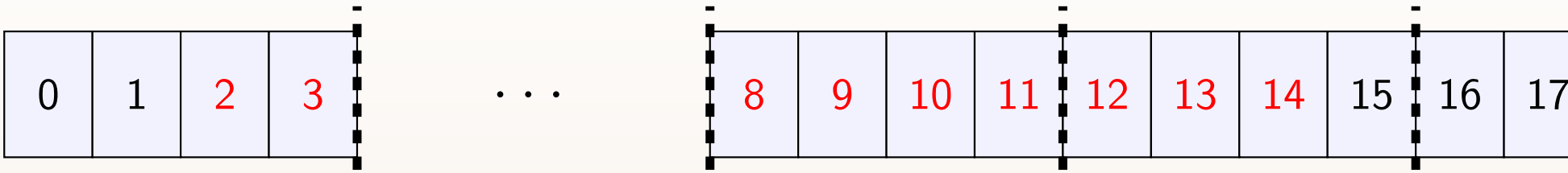
Με τη βοήθεια του A_{blocked}



Για κάθε ερώτηση $\mathrm{RMQ}(i, j)$ γράφω $[i, j] = I_1 \cup I_2 \cup I_3$ ώστε

1. Το I_2 είναι της μορφής $[i' \cdot B, j' \cdot B]$
2. $I_1 = [i + 1, i' \cdot B]$ και είναι μήκους $\leq B$.

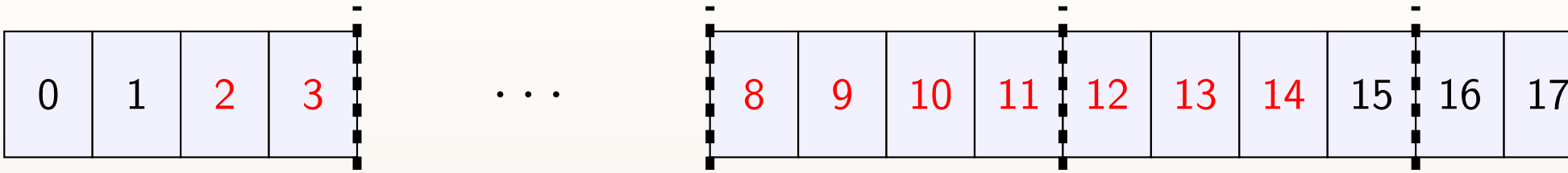
Με τη βοήθεια του A_{blocked}



Για κάθε ερώτηση $\mathrm{RMQ}(i, j)$ γράφω $[i, j] = I_1 \cup I_2 \cup I_3$ ώστε

1. Το I_2 είναι της μορφής $[i' \cdot B, j' \cdot B]$
2. $I_1 = [i + 1, i' \cdot B]$ και είναι μήκους $\leq B$.
3. $J_1 = [j' \cdot B + 1, j]$ και είναι μήκους $\leq B$.

Με τη βοήθεια του A_{blocked}



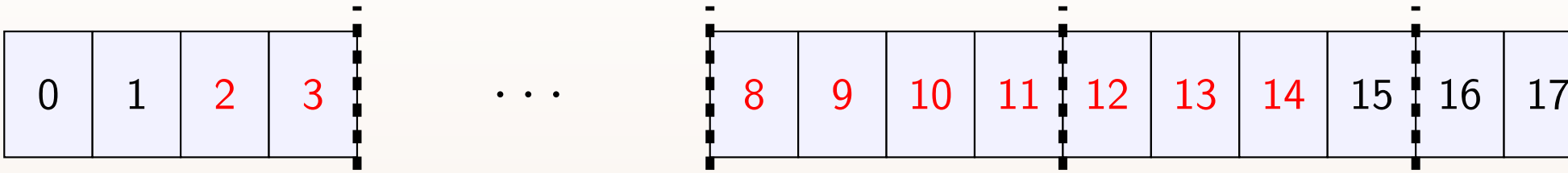
Για κάθε ερώτηση $\text{RMQ}(i, j)$ γράφω $[i, j] = I_1 \cup I_2 \cup I_3$ ώστε

1. Το I_2 είναι της μορφής $[i' \cdot B, j' \cdot B]$
2. $I_1 = [i + 1, i' \cdot B]$ και είναι μήκους $\leq B$.
3. $J_1 = [j' \cdot B + 1, j]$ και είναι μήκους $\leq B$.

Άρα αν βρω τα I_1, I_2, I_3 σε $O(1)$ χρόνο μπορώ τότε

1. $\min_{k \in I_1} A[k], \min_{k \in I_2} A[k]$ σε $O(B)$ χρόνο εφόσον $|I_1|, |I_2| \leq B$,

Με τη βοήθεια του A_{blocked}



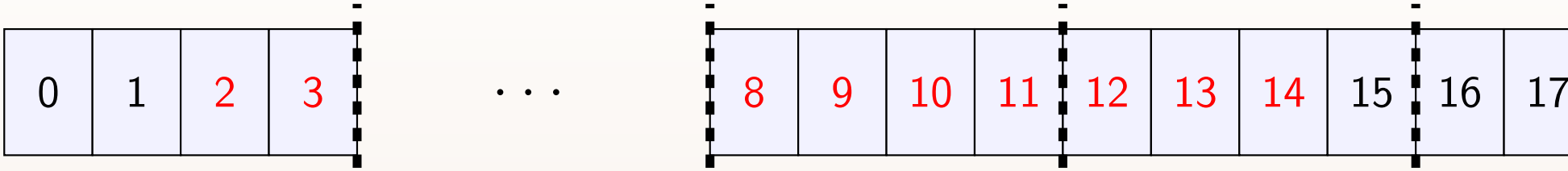
Για κάθε ερώτηση $\text{RMQ}(i, j)$ γράφω $[i, j] = I_1 \cup I_2 \cup I_3$ ώστε

1. Το I_2 είναι της μορφής $[i' \cdot B, j' \cdot B]$
2. $I_1 = [i + 1, i' \cdot B]$ και είναι μήκους $\leq B$.
3. $J_1 = [j' \cdot B + 1, j]$ και είναι μήκους $\leq B$.

Άρα αν βρω τα I_1, I_2, I_3 σε $O(1)$ χρόνο μπορώ τότε

1. $\min_{k \in I_1} A[k], \min_{k \in I_2} A[k]$ σε $O(B)$ χρόνο εφόσον $|I_1|, |I_2| \leq B$,
2. $\min_{k \in I_2} A[k] = \min_{k' \in [i', j']} \text{A}_{\text{blocked}}[k']$ σε $O(\frac{n}{B})$ χρόνο.

Με τη βοήθεια του A_{blocked}



Για κάθε ερώτηση $\text{RMQ}(i, j)$ γράφω $[i, j] = I_1 \cup I_2 \cup I_3$ ώστε

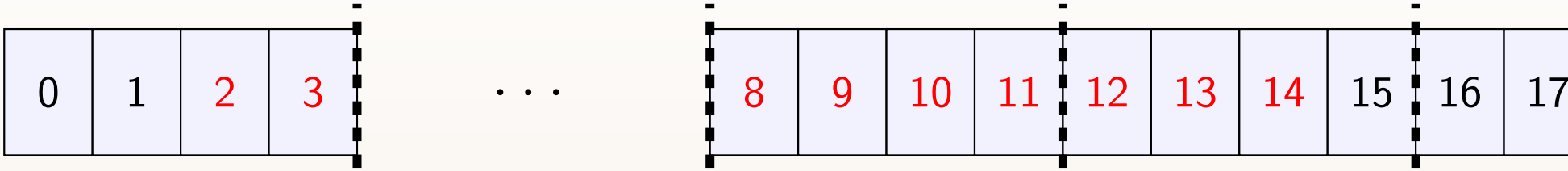
1. Το I_2 είναι της μορφής $[i' \cdot B, j' \cdot B]$
2. $I_1 = [i + 1, i' \cdot B]$ και είναι μήκους $\leq B$.
3. $J_1 = [j' \cdot B + 1, j]$ και είναι μήκους $\leq B$.

Άρα αν βρω τα I_1, I_2, I_3 σε $O(1)$ χρόνο μπορώ τότε

1. $\min_{k \in I_1} A[k], \min_{k \in I_2} A[k]$ σε $O(B)$ χρόνο εφόσον $|I_1|, |I_2| \leq B$,
2. $\min_{k \in I_2} A[k] = \min_{k' \in [i', j']} \text{A}_{\text{blocked}}[k']$ σε $O(\frac{n}{B})$ χρόνο.

Άρα μπορώ να απαντήσω όλη την ερώτηση σε $O\left(B + \frac{n}{B}\right)$ χρόνο

Με τη βοήθεια του A_{blocked}



Για κάθε ερώτηση $\text{RMQ}(i, j)$ γράφω $[i, j] = I_1 \cup I_2 \cup I_3$ ώστε

1. Το I_2 είναι της μορφής $[i' \cdot B, j' \cdot B]$
2. $I_1 = [i + 1, i' \cdot B]$ και είναι μήκους $\leq B$.
3. $J_1 = [j' \cdot B + 1, j]$ και είναι μήκους $\leq B$.

Άρα αν βρω τα I_1, I_2, I_3 σε $O(1)$ χρόνο μπορώ τότε

1. $\min_{k \in I_1} A[k], \min_{k \in I_2} A[k]$ σε $O(B)$ χρόνο εφόσον $|I_1|, |I_2| \leq B$,
2. $\min_{k \in I_2} A[k] = \min_{k' \in [i', j']} \text{A}_{\text{blocked}}[k']$ σε $O(\frac{n}{B})$ χρόνο.

Άρα μπορώ να απαντήσω όλη την ερώτηση σε $O\left(B + \frac{n}{B}\right)$ χρόνο

$$\Longrightarrow$$

διαλέγοντας $B = \lceil \sqrt{n} \rceil$ παίρνω $\langle n, \sqrt{n} \rangle$ χρόνο.

Μία πιο έξυπνη λύση.

Γιατί να μην κάνουμε το ίδιο με διαστήματα μήκους δυνάμεων του 2;

Μία πιο έξυπνη λύση.

Γιατί να μην κάνουμε το ίδιο με διαστήματα μήκους δυνάμεων του 2;

Ας προυπολογίσω τον $\widetilde{A}[i,\ell] = \min_{k\in[i,i+2^{\ell}-1]} A[k]$, ο οποίος έχει $O(n\log n)$ εγγραφές.

Μία πιο έξυπνη λύση.

Γιατί να μην κάνουμε το ίδιο με διαστήματα μήκους δυνάμεων του 2;

Ας προυπολογίσω τον $\widetilde{A}[i,\ell] = \min_{k\in[i,i+2^{\ell}-1]} A[k]$, ο οποίος έχει $O(n\log n)$ εγγραφές.

1. $\widetilde{A}[i,0] = A[i]$.

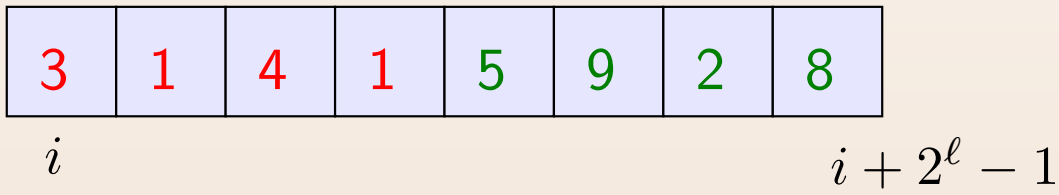
Μία πιο έξυπνη λύση.

Γιατί να μην κάνουμε το ίδιο με διαστήματα μήκους δυνάμεων του 2;

Ας προυπολογίσω τον $\widetilde{A}[i,\ell] = \min_{k\in[i,i+2^\ell-1]} A[k]$, ο οποίος έχει $O(n \log n)$ εγγραφές.

1. $\widetilde{A}[i,0] = A[i]$.
2. Παρατηρώ ότι κάθε διάστημα μήκους 2^ℓ με αριστερό άκρο το i μπορώ να το σπάσω σε δύο διαστήματα μήκους $2^{\ell-1}$:

$$\widetilde{A}[i,\ell] = \min\left\{\widetilde{A}[i,\ell-1], \widetilde{A}[i+2^\ell,\ell-1]\right\}$$



Απαντώντας μία ερώτηση.

Δοθέντος $[i, j]$ βρίσκουμε το μέγιστο ℓ ώστε $i + 2^{\ell} - 1 \leq j$, και παρατηρούμε ότι

Απαντώντας μία ερώτηση.

Δοθέντος $[i, j]$ βρίσκουμε το μέγιστο ℓ ώστε $i + 2^\ell - 1 \leq j$, και παρατηρούμε ότι

$$[i, j] = [i, i + 2^\ell - 1] \bigcup [j - 2^\ell + 1, j],$$

Απαντώντας μία ερώτηση.

Δοθέντος $[i, j]$ βρίσκουμε το μέγιστο ℓ ώστε $i + 2^\ell - 1 \leq j$, και παρατηρούμε ότι

$$[i, j] = [i, i + 2^\ell - 1] \bigcup [j - 2^\ell + 1, j],$$

άρα μπορούμε να απαντήσουμε την ερώτηση $\mathrm{RMQ}(i, j)$ ως

$$\min_{k \in [i,j]} A[k] = \min\{\widetilde{A}[i, \ell], \widetilde{A}[j - 2^\ell + 1, \ell]\}.$$

Απαντώντας μία ερώτηση.

Δοθέντος $[i, j]$ βρίσκουμε το μέγιστο ℓ ώστε $i + 2^\ell - 1 \leq j$, και παρατηρούμε ότι

$$[i, j] = [i, i + 2^\ell - 1] \bigcup [j - 2^\ell + 1, j],$$

άρα μπορούμε να απαντήσουμε την ερώτηση $\mathrm{RMQ}(i, j)$ ως

$$\min_{k \in [i,j]} A[k] = \min\{\widetilde{A}[i, \ell], \widetilde{A}[j - 2^\ell + 1, \ell]\}.$$

$\star$ Το ℓ το βρίσκουμε ως $\ell = \lfloor \log(j - i + 1) \rfloor$ (ή μπορούμε να προυπολογίσουμε όλα τα $\lfloor \log k \rfloor$ για $k \in [n]$.

Απαντώντας μία ερώτηση.

Δοθέντος $[i, j]$ βρίσκουμε το μέγιστο ℓ ώστε $i + 2^\ell - 1 \leq j$, και παρατηρούμε ότι

$$[i, j] = [i, i + 2^\ell - 1] \bigcup [j - 2^\ell + 1, j],$$

άρα μπορούμε να απαντήσουμε την ερώτηση $\mathrm{RMQ}(i, j)$ ως

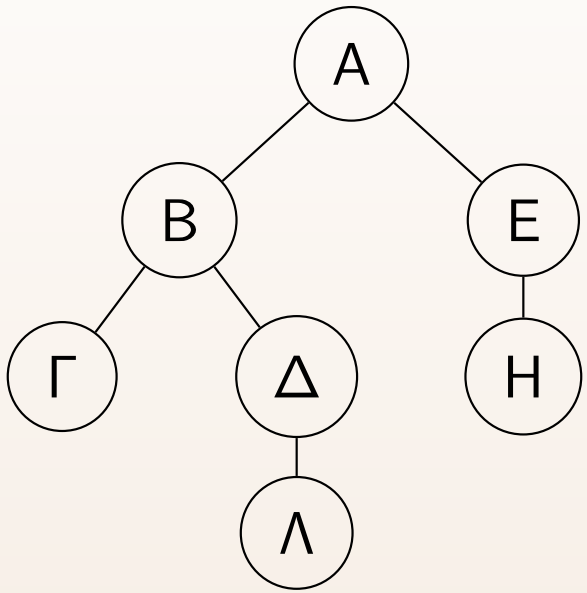
$$\min_{k \in [i,j]} A[k] = \min\{\widetilde{A}[i, \ell], \widetilde{A}[j - 2^\ell + 1, \ell]\}.$$

$\star$ Το ℓ το βρίσκουμε ως $\ell = \lfloor \log(j - i + 1) \rfloor$ (ή μπορούμε να προυπολογίσουμε όλα τα $\lfloor \log k \rfloor$ για $k \in [n]$.

Θεώρημα

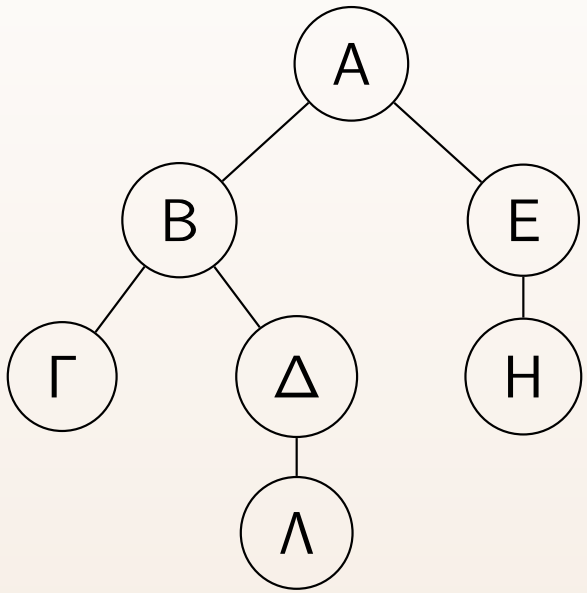
Υπάρχει μία δομή δεδομένων για το RMQ *με χρόνο* $\langle n \log n, 1 \rangle$.

Μία παρόμοια τακτική για το LCA.



Ας κρατήσω για κάθε κόμβο u τον πίνακα $P[u]$ ο οποίος αποθηκεύει τον πατέρα του u και έναν πίνακα $D[u]$ ο οποίος εκφράζει το βάθος του u στο δέντρο. Για τη ρίζα $P[r] = -1, D[r] = 0$

Μία παρόμοια τακτική για το LCA.



Ας κρατήσω για κάθε κόμβο u τον πίνακα $P[u]$ ο οποίος αποθηκεύει τον πατέρα του u και έναν πίνακα $D[u]$ ο οποίος εκφράζει το βάθος του u στο δέντρο. Για τη ρίζα $P[r] = -1, D[r] = 0$

Κάθε ερώτηση μπορώ να την απαντήσω σε χρόνο $\langle n, n \rangle$. Πώς;

Μία παρόμοια σχέση για το LCA.

Δημιουργούμε έναν βοηθητικό πίνακα $\widetilde{P}$ ο οποίος θα μας επιτρέπει να κάνουμε 'μεγάλα άλματα'.

Μία παρόμοια σχέση για το LCA.

Δημιουργούμε έναν βοηθητικό πίνακα $\widetilde{P}$ ο οποίος θα μας επιτρέπει να κάνουμε 'μεγάλα άλματα'.Έστω $\widetilde{P}[u][\ell]$ ο κόμβος που θα καταλήξει ο u αν κάνει 2^ℓ βήματα προς τα πάνω.

Μία παρόμοια σχέση για το LCA.

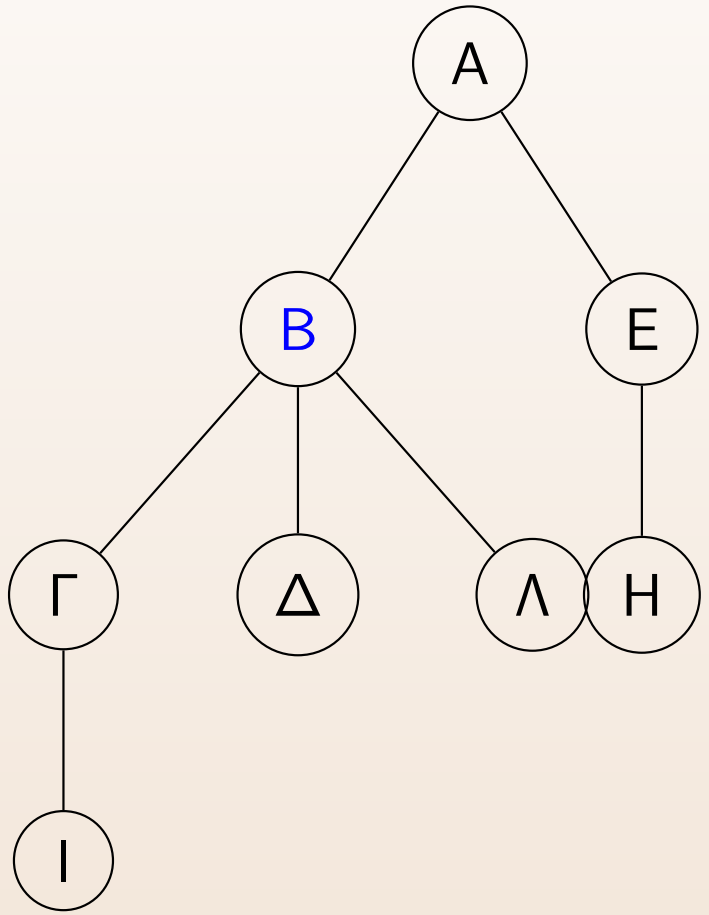
Δημιουργούμε έναν βοηθητικό πίνακα $\widetilde{P}$ ο οποίος θα μας επιτρέπει να κάνουμε 'μεγάλα άλματα'.Έστω $\widetilde{P}[u][\ell]$ ο κόμβος που θα καταλήξει ο u αν κάνει 2^ℓ βήματα προς τα πάνω.

Όταν $\ell = 0$ έχουμε $\widetilde{P}[u][0] = P[u]$.

Μία παρόμοια σχέση για το LCA.

Δημιουργούμε έναν βοηθητικό πίνακα $\widetilde{P}$ ο οποίος θα μας επιτρέπει να κάνουμε 'μεγάλα άλματα'.Έστω $\widetilde{P}[u][\ell]$ ο κόμβος που θα καταλήξει ο u αν κάνει 2^ℓ βήματα προς τα πάνω.

Όταν $\ell = 0$ έχουμε $\widetilde{P}[u][0] = P[u]$.
Για $\ell > 1$ παίρνουμε τη σχέση
$\widetilde{P}[u][\ell] = \widetilde{P}[\widetilde{P}[u][\ell-1]][\ell-1]$.



☛ Ο B είναι ο ενδιάμεσος κόμβος $\widetilde{P}[\Delta][0]$ στον υπολογισμό του $\widetilde{P}[\Delta][1]$.

⋆ Δυναμικός προγραμματισμός κατά αύξοντα ℓ σε χρόνο $O(n \log n)$.

Χρησιμοποιώντας τους πίνακες $\widetilde{P}, D$: Βήμα Ι.

Μπορεί ο u, v της ερώτησης μας να μην είναι στο ίδιο επίπεδο: το πρώτο βήμα είναι να τους φέρουμε στο ίδιο επίπεδο.

Αν $D[u] > D[v] \Rightarrow$ ο u πρέπει να ανέβει $D[u] - D[v]$ επίπεδα.

Χρησιμοποιώντας τους πίνακες $\widetilde{P}, D$: Βήμα Ι.

Μπορεί ο u, v της ερώτησης μας να μην είναι στο ίδιο επίπεδο: το πρώτο βήμα είναι να τους φέρουμε στο ίδιο επίπεδο.

Αν $D[u] > D[v] \Rightarrow$ ο u πρέπει να ανέβει $D[u] - D[v]$ επίπεδα.

$\star$ Γράφουμε τον αριθμό $D[u] - D[v]$ ως άθρισμα δυνάμεων του 2 και κάνουμε ένα αντίστοιχο άλμα για κάθε τέτοια δύναμη!

Χρησιμοποιώντας τους πίνακες $\widetilde{P}, D$: Βήμα Ι.

Μπορεί ο u, v της ερώτησης μας να μην είναι στο ίδιο επίπεδο: το πρώτο βήμα είναι να τους φέρουμε στο ίδιο επίπεδο.

Αν $D[u] > D[v] \Rightarrow$ ο u πρέπει να ανέβει $D[u] - D[v]$ επίπεδα.

$\star$ Γράφουμε τον αριθμό $D[u] - D[v]$ ως άθρισμα δυνάμεων του 2 και κάνουμε ένα αντίστοιχο άλμα για κάθε τέτοια δύναμη!

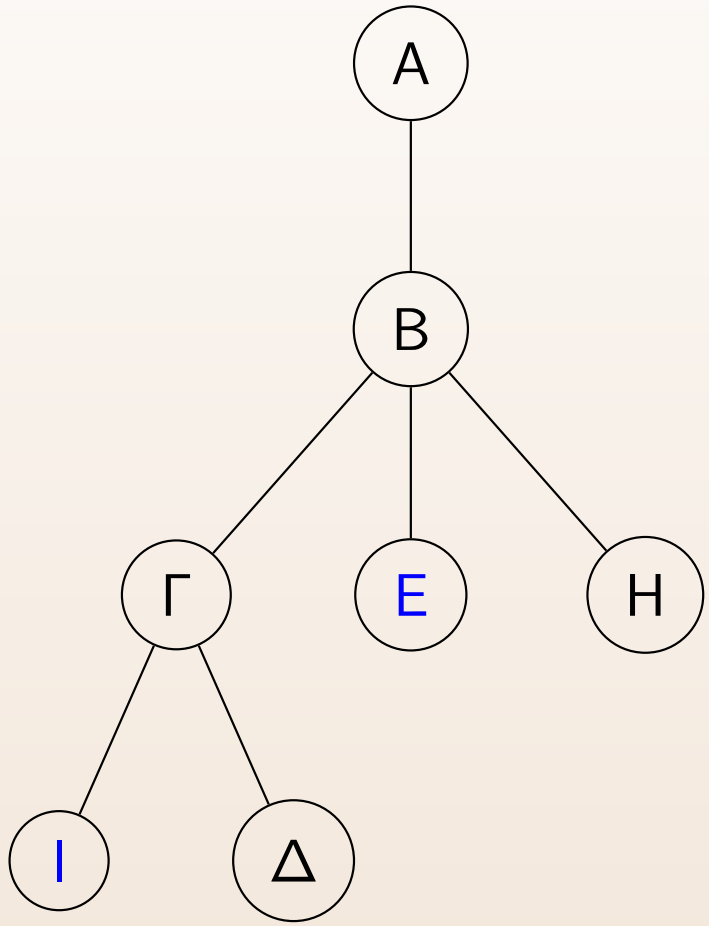
☛ Εύκολη υλοποίηση μέσω δυαδικής αναπαράστασης.

Χρησιμοποιώντας τους πίνακες $\widetilde{P}, D$: Βήμα Ι.

Μπορεί ο u, v της ερώτησης μας να μην είναι στο ίδιο επίπεδο: το πρώτο βήμα είναι να τους φέρουμε στο ίδιο επίπεδο.

Αν $D[u] > D[v] \Rightarrow$ ο u πρέπει να ανέβει $D[u] - D[v]$ επίπεδα.

$\star$ Γράφουμε τον αριθμό $D[u] - D[v]$ ως άθρισμα δυνάμεων του 2 και κάνουμε ένα αντίστοιχο άλμα για κάθε τέτοια δύναμη!



☛ Εύκολη υλοποίηση μέσω δυαδικής αναπαράστασης.
☛ $D[u] - D[v] = (110)_2 \Rightarrow$ $\widetilde{P}[\widetilde{P}[u][2]][1]$ είναι στο ίδιο ύψος με τον v.

Χρησιμοποιώντας τους πίνακες $\widetilde{P}, D$: Βήμα II.

Τώρα οι u, v είναι στο ίδιο ύψος, και θέλουμε να βρούμε τον ελάχιστο κοινό τους πρόγονο $w^\star$.

Χρησιμοποιώντας τους πίνακες $\widetilde{P}, D$: Βήμα II.

Τώρα οι u, v είναι στο ίδιο ύψος, και θέλουμε να βρούμε τον ελάχιστο κοινό τους πρόγονο $w^\star$.

👉 Μπορούμε να δοκιμάσουμε να κάνουμε άλματα μήκους 2^ℓ για κάποιο ℓ και να δούμε αν φτάσαμε στον ίδιο κόμβο.

Χρησιμοποιώντας τους πίνακες $\widetilde{P}, D$: Βήμα ΙΙ.

Τώρα οι u, v είναι στο ίδιο ύψος, και θέλουμε να βρούμε τον ελάχιστο κοινό τους πρόγονο $w^\star$.

☛ Μπορούμε να δοκιμάσουμε να κάνουμε άλματα μήκους 2^ℓ για κάποιο ℓ και να δούμε αν φτάσαμε στον ίδιο κόμβο.

Αλγόριθμος Εύρεσης LCA

1: Για ℓ από $\lfloor \log n \rfloor$ ως 0 κάνε
2: Αν $\widetilde{P}[u][\ell] \neq \widetilde{P}[v][\ell]$
3: $u = \widetilde{P}[u][\ell], v = \widetilde{P}[v][\ell]$
4: *//Εφόσον το άλμα μεγέθους 2^ℓ δεν φτάνει το $w^\star$, κάνε το*
5: Επέστρεψε το $P[u]$ ως κοινό πρόγονο.
6: *//Στο τέλος και οι δύο φτάσανε σε μία κατάσταση όπου ο πατέρας τους είναι το $w^\star$*

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ.

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

110100

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

$1\mathbf{1}0100 \Rightarrow 0\mathbf{1}0100$

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

$110100 \Rightarrow 010100 \Rightarrow 000100$

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

$110100 \Rightarrow 010100 \Rightarrow 000100 \Rightarrow 000100$

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

$110100 \Rightarrow 010100 \Rightarrow 000100 \Rightarrow 000100 \Rightarrow 00100$

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

$110100 \Rightarrow 010100 \Rightarrow 000100 \Rightarrow 000100 \Rightarrow 00100 \Rightarrow 00010$

Επεξήγηση του αλγόριθμου.

Αν ξέραμε το $D[w^\star]$ θα γράφαμε το $D[u] - D[w^\star]$ στο δυαδικό σύστημα και θα κάναμε τα αντίστοιχα άλματα.

Αντ' αυτού, εμείς επί της ουσίας πρέπει να μάθουμε το $D[w^\star]$. Έστω το $X := D[u] - D[w^\star]$ η απόσταση που πρέπει να διανύσουμε.

- Ο έλεγχος $\widetilde{P}[u][\ell] = \widetilde{P}[v][\ell]$ ρωτάει 'Ισχύει ότι $X \leq 2^\ell$;'
- Αν ναι, τότε ίσως να μην είναι ασφαλές να κάνουμε άλμα μεγέθους 2^ℓ. Αν όχι, τότε κάνουμε άλμα μεγέθους 2^ℓ, άρα έχουμε να καλύψουμε απόσταση ακόμη $X - 2^\ell$.
- Ουσιαστικά, ο αλγόριθμος βρίσκει τα bits του X από τα αριστερά προς τα δεξιά μέχρι ο αριθμός να μείνει με έναν άσσο στη δυαδική του αναπαράσταση: τότε πάντα ο X καταλήγει στη μονάδα.

$110100 \Rightarrow 010100 \Rightarrow 000100 \Rightarrow 000100 \Rightarrow 00100 \Rightarrow 00010 \Rightarrow 00001$

Συνοψίζοντας.

Θεώρημα

Υπάρχει δομή δεδομένων χρόνου $\langle n \log n, \log n \rangle$ *για το πρόβλημα* LCA.

Στο επόμενο μάθημα θα δούμε πως συνδέονται τα προβλήματα $\mathrm{RMQ}, \mathrm{LCA}$ και πως αυτή η σύνδεση δίνει δομή δεδομένων $\langle n, 1 \rangle$.