

Προηγμένα Θέματα Αλγορίθμων

Πιθανοτικοί Αλγόριθμοι I

Βασίλειος Νάκος

Τμήμα Πληροφορικής και Τηλεπικοινωνιών
Πανεπιστήμιο Αθηνών

Τι είναι ένας πιθανοτικός αλγόριθμος.

★ Ένας αλγόριθμος που παίρνει *τυχαίες* αποφάσεις.

Τι είναι ένας πιθανοτικός αλγόριθμος.

★ Ένας αλγόριθμος που παίρνει *τυχαίες* αποφάσεις.

Τι σημαίνει αυτό;

Τι είναι ένας πιθανοτικός αλγόριθμος.

★ Ένας αλγόριθμος που παίρνει *τυχαίες* αποφάσεις.

Τι σημαίνει αυτό;

Ένας αλγόριθμος $A(x, r)$ που πέρα από την είσοδο παίρνει μια συμβολοσειρά r και η εκτέλεσή του εξαρτάται από την r .

Τι είναι ένας πιθανοτικός αλγόριθμος.

★ Ένας αλγόριθμος που παίρνει *τυχαίες* αποφάσεις.

Τι σημαίνει αυτό;

Ένας αλγόριθμος $A(x, r)$ που πέρα από την είσοδο παίρνει μια συμβολοσειρά r και η εκτέλεσή του εξαρτάται από την r .

Μια *κατανομή* πάνω σε ντετερμινιστικούς αλγορίθμους.

Δύο είδη πιθανοτικών αλγορίθμων.

Έστω ότι κάποιος μας λέει ότι ένας πιθανοτικός αλγόριθμος τρέχει σε χρόνο $T(n)$. Τι μπορεί να σημαίνει αυτό;

Δύο είδη πιθανοτικών αλγορίθμων.

Έστω ότι κάποιος μας λέει ότι ένας πιθανοτικός αλγόριθμος τρέχει σε χρόνο $T(n)$. Τι μπορεί να σημαίνει αυτό;

- 1 Monte Carlo: ο αλγόριθμος τρέχει σωστά σε χρόνο $T(n)$ και δίνει σωστό αποτέλεσμα με πιθανότητα $\geq \frac{2}{3}$.

Δύο είδη πιθανοτικών αλγορίθμων.

Έστω ότι κάποιος μας λέει ότι ένας πιθανοτικός αλγόριθμος τρέχει σε χρόνο $T(n)$. Τι μπορεί να σημαίνει αυτό;

- 1 Monte Carlo: ο αλγόριθμος τρέχει σωστά σε χρόνο $T(n)$ και δίνει σωστό αποτέλεσμα με πιθανότητα $\geq \frac{2}{3}$.
- 2 Las Vegas: ο αλγόριθμος τρέχει πάντα σωστά και έχει αναμενόμενο χρόνο εκτέλεσης $T(n)$.

Έστω ένας πιθανοτικός Monte Carlo αλγόριθμος που παίρνει συμβολοσειρές μήκους R .

Monte Carlo αλγόριθμοι.

Έστω ένας πιθανοτικός Monte Carlo αλγόριθμος που παίρνει συμβολοσειρές μήκους R .

$$\forall x, \Pr_{r \sim \{0,1\}^R}[A(x, r) = \text{ΣΩΣΤΟ ΑΠΟΤΕΛΕΣΜΑ}] \geq \frac{2}{3}$$

👁 Τι σημαίνει αυτό; Πως αν πάρω ένα r στην τύχη θα πάρω το σωστό αποτέλεσμα με πιθανότητα $\geq \frac{2}{3}$. Ή με άλλα λόγια

👉 Για τουλάχιστον $2/3$ από τις πιθανές συμβολοσειρές ο αλγόριθμος τρέχει σωστά.

Έστω ένας πιθανοτικός Monte Carlo αλγόριθμος που παίρνει συμβολοσειρές μήκους R .

$$\forall x, \Pr_{r \sim \{0,1\}^R}[A(x, r) = \text{ΣΩΣΤΟ ΑΠΟΤΕΛΕΣΜΑ}] \geq \frac{2}{3}$$

👁 Τι σημαίνει αυτό; Πως αν πάρω ένα r στην τύχη θα πάρω το σωστό αποτέλεσμα με πιθανότητα $\geq \frac{2}{3}$. Ή με άλλα λόγια

Monte Carlo αλγόριθμοι.

Έστω ένας πιθανοτικός Monte Carlo αλγόριθμος που παίρνει συμβολοσειρές μήκους R .

$$\forall x, \Pr_{r \sim \{0,1\}^R}[A(x, r) = \text{ΣΩΣΤΟ ΑΠΟΤΕΛΕΣΜΑ}] \geq \frac{2}{3}$$

👁 Τι σημαίνει αυτό; Πως αν πάρω ένα r στην τύχη θα πάρω το σωστό αποτέλεσμα με πιθανότητα $\geq \frac{2}{3}$. Ή με άλλα λόγια

👉 Για τουλάχιστον $2/3$ από τις πιθανές συμβολοσειρές ο αλγόριθμος τρέχει σωστά.

Έστω ένας πιθανοτικός Monte Carlo αλγόριθμος που παίρνει συμβολοσειρές μήκους R .

$$\forall x, \Pr_{r \sim \{0,1\}^R} [A(x, r) = \text{ΣΩΣΤΟ ΑΠΟΤΕΛΕΣΜΑ}] \geq \frac{2}{3}$$

👁 Τι σημαίνει αυτό; Πως αν πάρω ένα r στην τύχη θα πάρω το σωστό αποτέλεσμα με πιθανότητα $\geq \frac{2}{3}$. Ή με άλλα λόγια

👉 Για τουλάχιστον $2/3$ από τις πιθανές συμβολοσειρές ο αλγόριθμος τρέχει σωστά.

Έστω μία τυχαία μεταβλητή X . Η αναμενόμενη τιμή της ισούται με

① $E[X] = \sum_{i=-\infty}^{\infty} \Pr[X = i] \cdot i$ εφόσον η X παίρνει τιμές στους ακεραίους,

Έστω μία τυχαία μεταβλητή X . Η αναμενόμενη τιμή της ισούται με

- 1 $E[X] = \sum_{i=-\infty}^{\infty} \Pr[X = i] \cdot i$ εφόσον η X παίρνει τιμές στους ακεραίους,
- 2 $E[X] = \sum_{i \in A} \Pr[X = i] \cdot i$ αν η X παίρνει τιμές σε ένα οποιοδήποτε αριθμήσιμο (ή πεπερασμένο) σύνολο.

Έστω μία τυχαία μεταβλητή X . Η αναμενόμενη τιμή της ισούται με

① $E[X] = \sum_{i=-\infty}^{\infty} \Pr[X = i] \cdot i$ εφόσον η X παίρνει τιμές στους ακεραίους,

② $E[X] = \sum_{i \in A} \Pr[X = i] \cdot i$ αν η X παίρνει τιμές σε ένα οποιοδήποτε αριθμήσιμο (ή πεπερασμένο) σύνολο.

👉 Ένας Las Vegas αλγόριθμος με χρόνο εκτέλεσης $T(n)$ σημαίνει:

Έστω μία τυχαία μεταβλητή X . Η αναμενόμενη τιμή της ισούται με

❶ $E[X] = \sum_{i=-\infty}^{\infty} \Pr[X = i] \cdot i$ εφόσον η X παίρνει τιμές στους ακεραίους,

❷ $E[X] = \sum_{i \in A} \Pr[X = i] \cdot i$ αν η X παίρνει τιμές σε ένα οποιοδήποτε αριθμήσιμο (ή πεπερασμένο) σύνολο.

👉 Ένας Las Vegas αλγόριθμος με χρόνο εκτέλεσης $T(n)$ σημαίνει: αν X η τυχαία μεταβλητή που αντιστοιχεί στον αναμενόμενο χρόνο εκτέλεσής του (για την χειρότερη είσοδο μήκος n) τότε $E[X] = T(n)$.

Η πιο σημαντική ιδιότητα της μέσης τιμής.

Αν X, Y τυχαίες μεταβλητές (πιθανώς εξαρτημένες) τότε $E[X + Y] = E[X] + E[Y]$.

Γραμμικότητα της Μέσης Τιμής.

Η πιο σημαντική ιδιότητα της μέσης τιμής.

Αν X, Y τυχαίες μεταβλητές (πιθανώς εξαρτημένες) τότε $E[X + Y] = E[X] + E[Y]$.

👉 Η ιδιότητα αυτή μας επιτρέπει να αναλύσουμε Las Vegas πιθανοτικούς αλγόριθμους,

Γραμμικότητα της Μέσης Τιμής.

Η πιο σημαντική ιδιότητα της μέσης τιμής.

Αν X, Y τυχαίες μεταβλητές (πιθανώς εξαρτημένες) τότε $E[X + Y] = E[X] + E[Y]$.

👉 Η ιδιότητα αυτή μας επιτρέπει να αναλύσουμε Las Vegas πιθανοτικούς αλγόριθμους, σπάζοντάς τους σε κομμάτια και αθροίζοντας τον αναμενόμενο χρόνο σε κάθε κομμάτι.

Η ανισότητα του Markov.

Έστω $X \geq 0$ τυχαία μεταβλητή. Τότε, $\Pr[X \geq \lambda \cdot E[X]] \leq \frac{1}{\lambda}$.

Η ανισότητα του Markov.

Έστω $X \geq 0$ τυχαία μεταβλητή. Τότε, $\Pr[X \geq \lambda \cdot E[X]] \leq \frac{1}{\lambda}$.

Απόδειξη: Έστω $\alpha := \lambda E[X]$. Πρώτα παρατηρούμε ότι

$$E[X] = \underbrace{\Pr[X < \alpha] \cdot E[X|X < \alpha]}_{\geq 0} + \Pr[X \geq \alpha] \cdot E[X|X > \alpha].$$

Η ανισότητα του Markov.

Έστω $X \geq 0$ τυχαία μεταβλητή. Τότε, $\Pr[X \geq \lambda \cdot E[X]] \leq \frac{1}{\lambda}$.

Απόδειξη: Έστω $\alpha := \lambda E[X]$. Πρώτα παρατηρούμε ότι

$$E[X] = \underbrace{\Pr[X < \alpha] \cdot E[X|X < \alpha]}_{\geq 0} + \Pr[X \geq \alpha] \cdot E[X|X > \alpha].$$

$$E[X] \geq \Pr[X \geq \alpha] \cdot E[X|X > \alpha]$$

Η ανισότητα του Markov.

Έστω $X \geq 0$ τυχαία μεταβλητή. Τότε, $\Pr[X \geq \lambda \cdot E[X]] \leq \frac{1}{\lambda}$.

Απόδειξη: Έστω $\alpha := \lambda E[X]$. Πρώτα παρατηρούμε ότι

$$E[X] = \underbrace{\Pr[X < \alpha] \cdot E[X|X < \alpha]}_{\geq 0} + \Pr[X \geq \alpha] \cdot E[X|X > \alpha].$$

$$E[X] \geq \Pr[X \geq \alpha] \cdot E[X|X > \alpha] \geq \Pr[X \geq \alpha] \cdot \alpha \checkmark$$

Las Vegas και Monte Carlo.

👁️ Αν έχω έναν Las Vegas αλγόριθμο τότε θα κατασκευάσω από αυτόν ένα Monte Carlo.

Las Vegas και Monte Carlo.

👁️ Αν έχω έναν Las Vegas αλγόριθμο τότε θα κατασκευάσω από αυτόν ένα Monte Carlo.

- ★ Monte Carlo $\Rightarrow$ τερματίζει σε χρόνο $T(n)$ και επιστρέφει σωστή απάντηση με πιθανότητα $\geq 2/3$
- ★ Las Vegas $\Rightarrow$ τερματίζει σε αναμενόμενο χρόνο $T(n)$ και επιστρέφει πάντα την σωστή απάντηση

Las Vegas και Monte Carlo.

👁️ Αν έχω έναν Las Vegas αλγόριθμο τότε θα κατασκευάσω από αυτόν ένα Monte Carlo.

- ★ Monte Carlo $\Rightarrow$ τερματίζει σε χρόνο $T(n)$ και επιστρέφει σωστή απάντηση με πιθανότητα $\geq 2/3$
- ★ Las Vegas $\Rightarrow$ τερματίζει σε αναμενόμενο χρόνο $T(n)$ και επιστρέφει πάντα την σωστή απάντηση

Προσομοίωση Monte Carlo: Τρέξε τον Las Vegas και αν δεν έχει τερματίσει σε $3 \cdot T(n)$ βήματα, σταματήσε τον.

Las Vegas και Monte Carlo.

👁️ Αν έχω έναν Las Vegas αλγόριθμο τότε θα κατασκευάσω από αυτόν ένα Monte Carlo.

- ★ Monte Carlo $\Rightarrow$ τερματίζει σε χρόνο $T(n)$ και επιστρέφει σωστή απάντηση με πιθανότητα $\geq 2/3$
- ★ Las Vegas $\Rightarrow$ τερματίζει σε αναμενόμενο χρόνο $T(n)$ και επιστρέφει πάντα την σωστή απάντηση

Προσομοίωση Monte Carlo: Τρέξε τον Las Vegas και αν δεν έχει τερματίσει σε $3 \cdot T(n)$ βήματα, σταματήσε τον.

👍 Αυτός είναι ένας Monte Carlo αλγόριθμος.

Las Vegas και Monte Carlo.

👁️ Αν έχω έναν Las Vegas αλγόριθμο τότε θα κατασκευάσω από αυτόν ένα Monte Carlo.

- ★ Monte Carlo $\Rightarrow$ τερματίζει σε χρόνο $T(n)$ και επιστρέφει σωστή απάντηση με πιθανότητα $\geq 2/3$
- ★ Las Vegas $\Rightarrow$ τερματίζει σε αναμενόμενο χρόνο $T(n)$ και επιστρέφει πάντα την σωστή απάντηση

Προσομοίωση Monte Carlo: Τρέξε τον Las Vegas και αν δεν έχει τερματίσει σε $3 \cdot T(n)$ βήματα, σταματήσε τον.

👉 Αυτός είναι ένας Monte Carlo αλγόριθμος.

★ Αν X ο χρόνος εκτέλεσης του Las Vegas τότε

$$\Pr[\text{ΛΑΘΟΣ ΑΠΑΝΤΗΣΗ ΤΟΥ MONTE CARLO}] = \Pr[X \geq 3 \cdot T(n)]$$

Las Vegas και Monte Carlo.

👁️ Αν έχω έναν Las Vegas αλγόριθμο τότε θα κατασκευάσω από αυτόν ένα Monte Carlo.

- ★ Monte Carlo $\Rightarrow$ τερματίζει σε χρόνο $T(n)$ και επιστρέφει σωστή απάντηση με πιθανότητα $\geq 2/3$
- ★ Las Vegas $\Rightarrow$ τερματίζει σε αναμενόμενο χρόνο $T(n)$ και επιστρέφει πάντα την σωστή απάντηση

Προσομοίωση Monte Carlo: Τρέξε τον Las Vegas και αν δεν έχει τερματίσει σε $3 \cdot T(n)$ βήματα, σταματήσε τον.

👍 Αυτός είναι ένας Monte Carlo αλγόριθμος.

★ Αν X ο χρόνος εκτέλεσης του Las Vegas τότε

$$\Pr[\text{ΛΑΘΟΣ ΑΠΑΝΤΗΣΗ ΤΟΥ MONTE CARLO}] = \Pr[X \geq 3 \cdot T(n)] \leq \frac{1}{3} \checkmark$$

Αλγόριθμος QuickSort

- 1: QuickSort(A) //πίνακας είσοδου
- 2: Διάλεξε στην τύχη $r \in [n]$
- 3: Έστω $A_L := \{a[i] < a[r]\}$ και $A_R := \{a[i] > a[r]\}$.
- 4: //Γραμμικός Χρόνος
- 5: $\bar{A}_L = \text{QuickSort}(A_L)$
- 6: $\bar{A}_R = \text{QuickSort}(A_R)$
- 7: Επέστρεψε τον ταξινομημένο A βάζοντας πρώτα $\bar{A}_L$ μετά $a[r]$, μετά $\bar{A}_R$.

Αναμενόμενος χρόνος εκτέλεσης: $O(n \log n)$

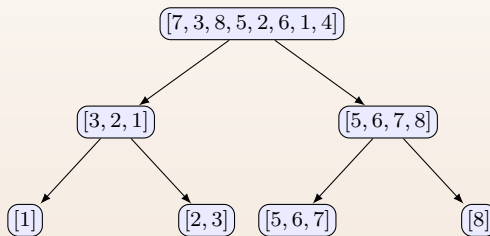
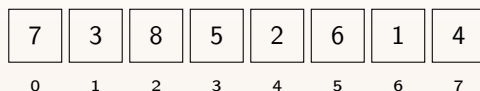
Αλγόριθμος QuickSort

- 1: QuickSort(A) //πίνακας είσοδου
- 2: Διάλεξε στην τύχη $r \in [n]$
- 3: Έστω $A_L := \{a[i] < a[r]\}$ και $A_R := \{a[i] > a[r]\}$.
- 4: //Γραμμικός Χρόνος
- 5: $\bar{A}_L = \text{QuickSort}(A_L)$
- 6: $\bar{A}_R = \text{QuickSort}(A_R)$
- 7: Επέστρεψε τον ταξινομημένο A βάζοντας πρώτα $\bar{A}_L$ μετά $a[r]$, μετά $\bar{A}_R$.

Αναμενόμενος χρόνος εκτέλεσης: $O(n \log n)$

Πώς γίνεται η ανάλυση;

Παράδειγμα των υποπροβλημάτων που προκύπτουν στην QuickSort.



👁 Αναμένουμε όρο κάθε υπο-προβλημα να κόπει περίπου στην μέση (λόγω τυχαιότητας).

Ανάλυση QuickSort I.

★ Έστω T_i ο αναμενόμενος χρόνος που τρέχει η QuickSort σε πίνακα μεγέθους i .

Ανάλυση QuickSort I.

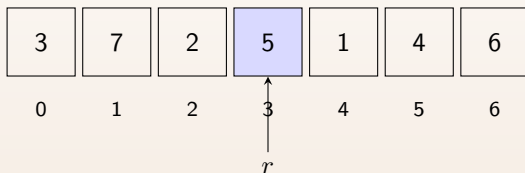
★ Έστω T_i ο αναμενόμενος χρόνος που τρέχει η QuickSort σε πίνακα μεγέθους i .

👁 Με πιθανότητα $\frac{1}{n}$ το $\bar{A}_L$ θα έχει i στοιχεία και ο $\bar{A}_R$ θα έχει $n - 1 - i$ στοιχεία.

Ανάλυση QuickSort I.

★ Έστω T_i ο αναμενόμενος χρόνος που τρέχει η QuickSort σε πίνακα μεγέθους i .

👁 Με πιθανότητα $\frac{1}{n}$ το $\bar{A}_L$ θα έχει i στοιχεία και ο $\bar{A}_R$ θα έχει $n - 1 - i$ στοιχεία.

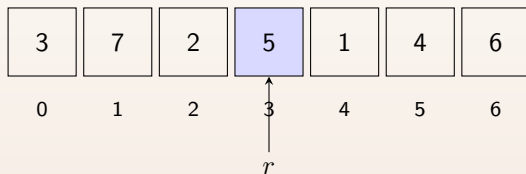


$$T_n = O(n) + \sum_{i=0}^{n-1} \frac{1}{n} \cdot (T(i) + T(n - 1 - i))$$

Ανάλυση QuickSort I.

★ Έστω T_i ο αναμενόμενος χρόνος που τρέχει η QuickSort σε πίνακα μεγέθους i .

👁 Με πιθανότητα $\frac{1}{n}$ το $\bar{A}_L$ θα έχει i στοιχεία και ο $\bar{A}_R$ θα έχει $n - 1 - i$ στοιχεία.

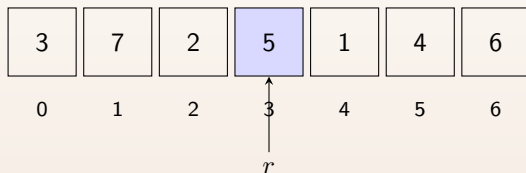


$$T_n = O(n) + \sum_{i=0}^{n-1} \frac{1}{n} \cdot (T(i) + T(n-1-i)) = \frac{2}{n} \sum_{i=0}^{n-1} T(i)$$

Ανάλυση QuickSort I.

★ Έστω T_i ο αναμενόμενος χρόνος που τρέχει η QuickSort σε πίνακα μεγέθους i .

👁 Με πιθανότητα $\frac{1}{n}$ το $\bar{A}_L$ θα έχει i στοιχεία και ο $\bar{A}_R$ θα έχει $n - 1 - i$ στοιχεία.



$$T_n = O(n) + \sum_{i=0}^{n-1} \frac{1}{n} \cdot (T(i) + T(n-1-i)) = \frac{2}{n} \sum_{i=0}^{n-1} T(i) = O(n \log n)$$

Ανάλυση Quicksort II.

👁 Μπορούμε να χρεώσουμε τον χρόνο εκτέλεσης του αλγορίθμου στις συγκρίσεις που έκανε!

Ανάλυση Quicksort II.

👁 Μπορούμε να χρεώσουμε τον χρόνο εκτέλεσης του αλγορίθμου στις συγκρίσεις που έκανε!

👉 $T_{i,j} = 1$ αν συγκρίθηκαν τα $a[i]$ και $a[j]$ μέσα στον αλγόριθμο.

Ανάλυση Quicksort II.

👁 Μπορούμε να χρεώσουμε τον χρόνο εκτέλεσης του αλγορίθμου στις συγκρίσεις που έκανε!

👉 $T_{i,j} = 1$ αν συγκρίθηκαν τα $a[i]$ και $a[j]$ μέσα στον αλγόριθμο.

Τότε $X := (\text{χρόνος αλγορίθμου}) = \sum_{i=1}^n \sum_{j=1}^n T_{i,j}$.

Ανάλυση Quicksort II.

👁 Μπορούμε να χρεώσουμε τον χρόνο εκτέλεσης του αλγορίθμου στις συγκρίσεις που έκανε!

👉 $T_{i,j} = 1$ αν συγκρίθηκαν τα $a[i]$ και $a[j]$ μέσα στον αλγόριθμο.

Τότε $X := (\text{χρόνος αλγορίθμου}) = \sum_{i=1}^n \sum_{j=1}^n T_{i,j}$.



👁 Για να συγκριθεί το $a[i]$ με το $a[j]$ πρέπει σε ένα από τα υποπροβλήματα που δημιουργούνται το $a[i]$ και το $a[j]$ να είναι στο ίδιο και ένα από τα δύο να τα διαλέξουμε για να σπάσουμε τον πίνακα.

Ανάλυση Quicksort II.

👁 Μπορούμε να χρεώσουμε τον χρόνο εκτέλεσης του αλγορίθμου στις συγκρίσεις που έκανε!

👉 $T_{i,j} = 1$ αν συγκρίθηκαν τα $a[i]$ και $a[j]$ μέσα στον αλγόριθμο.

Τότε $X := (\text{χρόνος αλγορίθμου}) = \sum_{i=1}^n \sum_{j=1}^n T_{i,j}$.



👁 Για να συγκριθεί το $a[i]$ με το $a[j]$ πρέπει σε ένα από τα υποπροβλήματα που δημιουργούνται το $a[i]$ και το $a[j]$ να είναι στο ίδιο και ένα από τα δύο να τα διαλέξουμε για να σπάσουμε τον πίνακα.

Ανάλυση Quicksort II.

👁 Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



Ανάλυση Quicksort II.

👁 Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



👉 $\Pr[T_{i,j} = 1] = \frac{2}{j-i+1}$ (γιατί;)

Ανάλυση Quicksort II.

👁️ Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



👉 $\Pr[T_{i,j} = 1] = \frac{2}{j-i+1}$ (γιατί;)

$$E[T_{i,j}] = \Pr[T_{i,j} = 1] \cdot 1 + \Pr[T_{i,j} = 0] \cdot 0 =$$

Ανάλυση Quicksort II.

👁️ Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



👉 $\Pr[T_{i,j} = 1] = \frac{2}{j-i+1}$ (γιατί;)

$$E[T_{i,j}] = \Pr[T_{i,j}] \cdot 1 + \Pr[T_{i,j} = 0] \cdot 0 = \frac{2}{j-i+1}$$

Ανάλυση Quicksort II.

👁️ Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



👉 $\Pr[T_{i,j} = 1] = \frac{2}{j-i+1}$ (γιατί;)

$$E[T_{i,j}] = \Pr[T_{i,j}] \cdot 1 + \Pr[T_{i,j} = 0] \cdot 0 = \frac{2}{j-i+1}$$

$$E[T] = E \left[\sum_{i,j} T_{i,j} \right]$$

Ανάλυση Quicksort II.

👁️ Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



👉 $\Pr[T_{i,j} = 1] = \frac{2}{j-i+1}$ (γιατί;)

$$E[T_{i,j}] = \Pr[T_{i,j}] \cdot 1 + \Pr[T_{i,j} = 0] \cdot 0 = \frac{2}{j-i+1}$$

$$E[T] = E \left[\sum_{i,j} T_{i,j} \right] = \sum_{i,j} E[T_{i,j}] =$$

Ανάλυση Quicksort II.

👁️ Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



👉 $\Pr[T_{i,j} = 1] = \frac{2}{j-i+1}$ (γιατί;)

$$E[T_{i,j}] = \Pr[T_{i,j} = 1] \cdot 1 + \Pr[T_{i,j} = 0] \cdot 0 = \frac{2}{j-i+1}$$

$$E[T] = E\left[\sum_{i,j} T_{i,j}\right] = \sum_{i,j} E[T_{i,j}] = \sum_{i,j} \frac{2}{j-i+1}$$

Ανάλυση Quicksort II.

👁️ Ας κοιτάξουμε την στιγμή που τα $a[i], a[j]$ χωρίστηκαν. Ποια είναι η πιθανότητα τότε να συγκρίθηκαν;



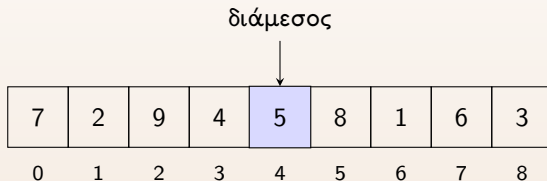
👉 $\Pr[T_{i,j} = 1] = \frac{2}{j-i+1}$ (γιατί;)

$$E[T_{i,j}] = \Pr[T_{i,j}] \cdot 1 + \Pr[T_{i,j} = 0] \cdot 0 = \frac{2}{j-i+1}$$

$$E[T] = E \left[\sum_{i,j} T_{i,j} \right] = \sum_{i,j} E[T_{i,j}] = \sum_{i,j} \frac{2}{j-i+1} \leq 2n \ln n \checkmark$$

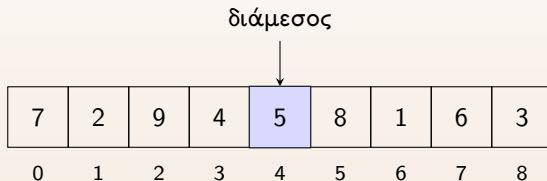
Πιθανοτικός αλγόριθμος για διάμεσο.

Δοθέντος ενός πίνακα A , μπορούμε να βρούμε την διάμέσό του χωρίς να ταξινομήσουμε τον πίνακα;



Πιθανοτικός αλγόριθμος για διάμεσο.

Δοθέντος ενός πίνακα A , μπορούμε να βρούμε την διάμέσό του χωρίς να ταξινομήσουμε τον πίνακα;



Πάρε ένα $a[r]$ στην τύχη. Τι πληροφορία μπορείς να πάρεις από αυτό;

Πιθανοτικός Αλγόριθμος για Διάμεσο.

★ Κάθε φορά παίρνω ένα στοιχείο $a[r]$ στην τύχη και χωρίζω τον πίνακα σε δύο υποπίνακες $A_L := \{i : a[i] < a[r]\}$ και $A_R := \{i : a[i] > a[r]\}$.

Πιθανοτικός Αλγόριθμος για Διάμεσο.

★ Κάθε φορά παίρνω ένα στοιχείο $a[r]$ στην τύχη και χωρίζω τον πίνακα σε δύο υποπίνακες $A_L := \{i : a[i] < a[r]\}$ και $A_R := \{i : a[i] > a[r]\}$.

- 1 Αν $|A_L| = |A_R|$ τότε r είναι η διάμεσος.
- 2 Αν $|A_L| < |A_R|$ τότε η διάμεσος του A είναι στον A_R το $\frac{n}{2} - k$ στοιχείο.

Πιθανοτικός Αλγόριθμος για Διάμεσο.

★ Κάθε φορά παίρνω ένα στοιχείο $a[r]$ στην τύχη και χωρίζω τον πίνακα σε δύο υποπίνακες $A_L := \{i : a[i] < a[r]\}$ και $A_R := \{i : a[i] > a[r]\}$.

- 1 Αν $|A_L| = |A_R|$ τότε r είναι η διάμεσος.
- 2 Αν $|A_L| < |A_R|$ τότε η διάμεσος του A είναι στον A_R το $\frac{n}{2} - k$ στοιχείο.
- 3 Αν $|A_L| > |A_R|$ τότε η διάμεσος του A είναι στον A_L το $\frac{n}{2}$ στοιχείο.

Πιθανοτικός Αλγόριθμος για Διάμεσο.

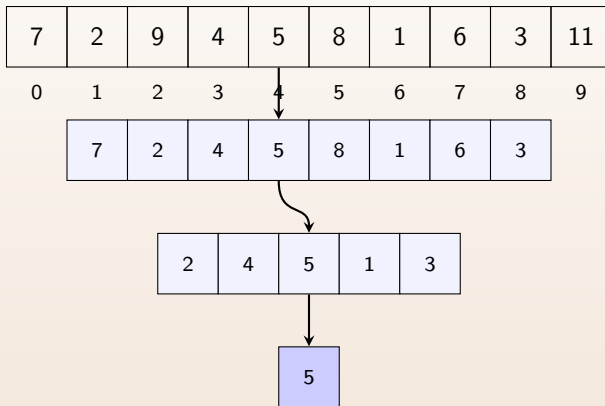
★ Κάθε φορά παίρνω ένα στοιχείο $a[r]$ στην τύχη και χωρίζω τον πίνακα σε δύο υποπίνακες $A_L := \{i : a[i] < a[r]\}$ και $A_R := \{i : a[i] > a[r]\}$.

- 1 Αν $|A_L| = |A_R|$ τότε r είναι η διάμεσος.
- 2 Αν $|A_L| < |A_R|$ τότε η διάμεσος του A είναι στον A_R το $\frac{n}{2} - k$ στοιχείο.
- 3 Αν $|A_L| > |A_R|$ τότε η διάμεσος του A είναι στον A_L το $\frac{n}{2}$ στοιχείο.

👁 Άρα κάθε φορά πρέπει να λύσω ένα πιο γενικό πρόβλημα: βρες μου το k -στό στοιχείο σε έναν πίνακα με n θέσεις σε χρόνο $O(n)$.

Πιθανοτικός αλγόριθμος για διάμεσο.

👉 Μπορούμε να απομονώσουμε το κομμάτι του πίνακα όπου ζει η διάμεσος και μετά να ψάξουμε το κατάλληλο στοιχείο σε εκείνο το κομμάτι.



👁 Κάθε φορά που προχωράμε σε έναν υποπίνακα, το μέγεθος του υποπίνακα εξαρτάται και από το k . Ας είμαστε απαισιοδόξοι και ας πάρουμε πάντα τον μεγαλύτερο από τους δύο υποπίνακες.

👁 Κάθε φορά που προχωράμε σε έναν υποπίνακα, το μέγεθος του υποπίνακα εξαρτάται και από το k . Ας είμαστε απαισιόδοξοι και ας πάρουμε πάντα τον μεγαλύτερο από τους δύο υποπίνακες.

★ Αν ο υποπίνακας με στοιχεία $< a[r]$ έχει μέγεθος ℓ ο άλλος υποπίνακας έχει μέγεθος $(n - 1 - \ell)$,

👁 Κάθε φορά που προχωράμε σε έναν υποπίνακα, το μέγεθος του υποπίνακα εξαρτάται και από το k . Ας είμαστε απαισιόδοξοι και ας πάρουμε πάντα τον μεγαλύτερο από τους δύο υποπίνακες.

★ Αν ο υποπίνακας με στοιχεία $< a[r]$ έχει μέγεθος ℓ ο άλλος υποπίνακας έχει μέγεθος $(n - 1 - \ell)$, άρα θα μείνουμε με έναν υποπίνακα μεγέθους $\max\{\ell, n - 1 - \ell\}$.

👁 Κάθε φορά που προχωράμε σε έναν υποπίνακα, το μέγεθος του υποπίνακα εξαρτάται και από το k . Ας είμαστε απαισιόδοξοι και ας πάρουμε πάντα τον μεγαλύτερο από τους δύο υποπίνακες.

★ Αν ο υποπίνακας με στοιχεία $< a[r]$ έχει μέγεθος ℓ ο άλλος υποπίνακας έχει μέγεθος $(n - 1 - \ell)$, άρα θα μείνουμε με έναν υποπίνακα μεγέθους $\max\{\ell, n - 1 - \ell\}$.

Άρα $\Pr_{\ell \sim [n]}[\max\{\ell, n - 1 - \ell\} = i] = \frac{2}{n}$ για $i \in [\frac{n}{2}, \dots, n - 1]$.

👁 Κάθε φορά που προχωράμε σε έναν υποπίνακα, το μέγεθος του υποπίνακα εξαρτάται και από το k . Ας είμαστε απαισιοδόξοι και ας πάρουμε πάντα τον μεγαλύτερο από τους δύο υποπίνακες.

★ Αν ο υποπίνακας με στοιχεία $< a[r]$ έχει μέγεθος ℓ ο άλλος υποπίνακας έχει μέγεθος $(n - 1 - \ell)$, άρα θα μείνουμε με έναν υποπίνακα μεγέθους $\max\{\ell, n - 1 - \ell\}$.

Άρα $\Pr_{\ell \sim [n]}[\max\{\ell, n - 1 - \ell\} = i] = \frac{2}{n}$ για $i \in [\frac{n}{2}, \dots, n - 1]$.

Αναδρομική Σχέση για τον αναμενόμενο χρόνο

εκτέλεσης $\Rightarrow T(n) \leq \frac{2}{n} \cdot (T(\frac{n}{2}) + T(\frac{n}{2} + 1) + \dots + T(n - 1))$

👁 Κάθε φορά που προχωράμε σε έναν υποπίνακα, το μέγεθος του υποπίνακα εξαρτάται και από το k . Ας είμαστε απαισιοδόξοι και ας πάρουμε πάντα τον μεγαλύτερο από τους δύο υποπίνακες.

★ Αν ο υποπίνακας με στοιχεία $< a[r]$ έχει μέγεθος ℓ ο άλλος υποπίνακας έχει μέγεθος $(n - 1 - \ell)$, άρα θα μείνουμε με έναν υποπίνακα μεγέθους $\max\{\ell, n - 1 - \ell\}$.

Άρα $\Pr_{\ell \sim [n]}[\max\{\ell, n - 1 - \ell\} = i] = \frac{2}{n}$ για $i \in [\frac{n}{2}, \dots, n - 1]$.

Αναδρομική Σχέση για τον αναμενόμενο χρόνο

εκτέλεσης $\Rightarrow T(n) \leq \frac{2}{n} \cdot (T(\frac{n}{2}) + T(\frac{n}{2} + 1) + \dots + T(n - 1)) = O(n)$.