Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Δημήτρης Φωτάκης, Δώρα Σούλιου, Θανάσης Λιανέας
1η Σειρά Γραπτών Ασκήσεων - Σχέδιο Λύσεων

1 Άσκηση 1 - Ασυμπτωτικός Συμβολισμός, Αναδρομικές Σχέσεις

(α): Ασυμπτωτικός Συμβολισμός

Έχουμε πως:

$$\sum_{k=1}^{n} k2^{-k} = \Theta(1)$$

διότι $\sum_{k=1}^{n} k2^{-k} \leq \sum_{k=1}^{\infty} k2^{-k} = 2$.

Έπειτα, είναι: $n2^{2^{2^{100}}} = \Theta(n)$, όπως και $\log(\binom{2n}{n}) = \Theta(n)$, διότι

$$2^n \leq \binom{2n}{n} \leq (2e)^n$$

Έχουμε $2^{(\log_2 \log_2 n)^4} = \Theta((\log_2 n)^{(\log_2 n)^3})$.

Επίσης, αφού $\log(n!) = \Theta(n \log n)$, είναι: $(\sqrt{n})^{\log_2 \log_2 (n!)} = O((\sqrt{n})^{\log_2 n})$.

Είναι: $n \sum_{k=0}^{n} \binom{n}{k} = \Theta(n2^n) = \sum_{k=1}^{n} k2^k$.

Απομένει η $\frac{\log(n!)}{(\log \log n)^5} = \frac{n \log(n)}{(\log \log n)^5}$

Άρα, για να τα συγκρίνω, λόγω μονοτονίας, αρκεί να ελέγχω τους λογαρίθμους τους.

Άσκηση 1 (β):

1. $T(n) = 6T(n/3) + n^2 \log n = \Theta(n^2 \log n)$ (MT)

2. $T(n) = 9T(n/3) + n^2 \log n = \Theta(n^2 \log^2 n)$ (Όχι MT, όχι πολυωνυμικά διαχωρίσιμες. Επειδή $n^{\log_3 9} = n^2$ και $f(n) = n^2 \log n$, υποπτευόμαστε $T(n) = \Theta(n^2 \log^2 n)$ και το αποδεικνύουμε με επαγωγή ή χρησιμοποιώντας το δέντρο αναδρομής)

3. $T(n) = 11T(n/3) + n^2 \log n = \Theta(n^{\log_3 11})$ (MT)

4. $T(n) = T(n/4) + T(n/2) + n = \Theta(n)$ (Επειδή $n/4 + n/2 < n$, υποπτευόμαστε $T(n) = \Theta(n)$ και το αποδεικνύουμε με επαγωγή ή χρησιμοποιώντας το δέντρο αναδρομής)

5. $T(n) = 2T(n/4) + T(n/2) + n = \Theta(n \log n)$ (Επειδή $2n/4 + n/2 = n$, υποπτευόμαστε $T(n) = \Theta(n \log n)$ και το αποδεικνύουμε με επαγωγή ή χρησιμοποιώντας το δέντρο αναδρομής)

6. $T(n) = T(n^{2/3}) + \Theta(\log n)$ (Είναι $T(n) \leq \Theta(\log n) \sum_{k=0}^{+\infty} (2/3)^k = \Theta(\log n)$)

7. $T(n) = T(n/3) + \sqrt{n} = \Theta(\sqrt{n})$ (Είναι $T(n) \leq \sqrt{n} \sum_{k=0}^{+\infty} (1/\sqrt{3})^k = \Theta(\sqrt{n})$)

Άσκηση 2 - Προθεματική Ταξινόμηση

(α)
- Βρίσκουμε το μεγαλύτερο στοιχείο του A.
- Με μία προθεματική περιστροφή, το φέρνουμε στην αριστερότερη θέση.
- Με μία ακόμη προθεματική περιστροφή, το μετακινούμε στην δεξιότερη θέση.
- Συνεχίζουμε αναδρομικά στον υποπίνακα $A[1..(n-1)]$.
- Μπορούμε να μειώσουμε περαιτέρω το πλήθος των κινήσεων, κατά έναν μικρό σταθερό όρο, αν θεωρήσουμε π.χ. ως βάση της αναδρομής το $n = 3$ και εξετάσουμε εξαντλητικά την περίπτωση αυτή (σε αυτή την περίπτωση, έχουμε ταξινόμηση με 3 κινήσεις).

(β) Στον αλγόριθμο του προηγούμενου ερωτήματος, προσθέτουμε το πολύ μία προθεματική περιστροφή του εκάστοτε μεγαλύτερου στοιχείου όταν αυτό βρίσκεται στην αριστερότερη θέση.

(γ)
1. Θεωρούμε ότι ο τρέχων πίνακας δεν έχει συμβατά ζεύγη, γιατί αν είχε κάποιο, θα μπορούσαμε να το αντικαταστήσουμε με ένα στοιχείο (του οποίου το πρόσημο ταυτίζεται με το κοινό πρόσημο των στοιχείων του ζεύγους) και να καταλήξουμε σε ένα "μικρότερο" ισοδύναμο πρόβλημα. Διακρίνουμε δύο περιπτώσεις.

   Αν υπάρχει στοιχείο με θετικό πρόσημο:

   * Βρίσκουμε το μεγαλύτερο (σε απόλυτη τιμή) θετικό στοιχείο, έστω x.
   * Αν $x = n$, με δύο κινήσεις (βλ. ερώτημα (α)) δημιουργούμε το τετριμμένο συμβατό ζεύγος.
   * Αλλιώς, το $x + 1$ έχει αρνητικό πρόσημο και έτσι μπορούμε να δημιουργήσουμε ένα νέο συμβατό ζεύγος με δύο κινήσεις το πολύ:
     - Αν το $-(x+1)$ βρίσκεται δεξιότερα του x, τότε κάνουμε μία προσημασμένη προθεματική περιστροφή που περιλαμβάνει μέχρι και το $-(x+1)$. Έτσι, έχουμε έναν πίνακα της μορφής: $[(x+1), \ldots, -x, \ldots]$. Με μία κίνηση, σχηματίζουμε το νέο συμβατό ζεύγος $(-(x+1), -x)$.
     - Διαφορετικά, η πρώτη μας κίνηση περιλαμβάνει μέχρι και το x (προκύπτει πίνακας της μορφής $[-x, \ldots, (x+1), \ldots]$) και η δεύτερη σχηματίζει το νέο συμβατό ζεύγος $(x, x+1)$.

   Αν όλα τα στοιχεία έχουν αρνητικό πρόσημο:

   * Από υπόθεση, δεν έχουμε τον πίνακα $[-1, -2, \ldots, -n]$.
   * Άρα, για κάποιο στοιχείο $-x$ ($x < n$), το $-(x+1)$ είναι αριστερότερα.
   * Με δύο κινήσεις ($[\ldots, -(x+1), \ldots, -x, \ldots] \rightarrow [(x+1), \ldots, -x, \ldots] \rightarrow [\ldots, -(x+1), -x, \ldots]$), δημιουργούμε το νέο συμβατό ζεύγος $(-(x+1), -x)$.
2. Ο συνολικός μας αλγόριθμος έχει τα ακόλουθα βήματα:

1. Απαλείφουμε όλα τα συμβατά ζεύγη αναδρομικά, αντικαθιστώντας τα με ένα προσημασμένο στοιχείο. Προκύπτει ισοδύναμο πρόβλημα, με μικρότερο πλήθος στοιχείων, έστω n'. Αν $n' = 0$, ο πίνακας έχει ταξινομηθεί.
2. Αν ο πίνακας είναι ο $[-1, -2, \ldots, -n']$, τότε:
   Επαναλαμβάνουμε για n' φορές:
   * Εκτέλεσε μία προσημασμένη προθεματική περιστροφή όλων των στοιχείων.
   * Εκτέλεσε μία προσημασμένη προθεματική περιστροφή των $n' - 1$ αριστερότερων στοιχείων.

   Ο πίνακας έχει ταξινομηθεί.
3. Αλλιώς, με βάση το ερώτημα 1, κατασκευάζουμε ένα νέο συμβατό ζεύγος και πηγαίνουμε στο βήμα 1 του αλγορίθμου.

Αρκεί να κατασκευάσουμε n συμβατά ζεύγη. Αν k το πλήθος επαναλήψεων του συνολικού αλγορίθμου, τότε το συνολικό πλήθος των κινήσεών μας είναι το πολύ: $2k + 2(n-k) = 2n$, γιατί σε κάθε επανάληψη (εκτός ίσως της τελευταίας) κάνουμε το πολύ δύο κινήσεις και κάθε μία από αυτές μειώνει το πλήθος των ζευγών που μας μένουν τουλάχιστον κατά 1. Στην τελευταία επανάληψη, πιθανώς να χρειαστεί να κάνουμε $2n' \leq 2(n-k)$ κινήσεις.

Άσκηση 3 - Παίζοντας Χαρτιά

Είσοδος: 6 3 5 2 4 1 (κορυφή τράπουλας 6)

<u>Round 1:</u> Το 6 εκκινεί το παιχνίδι και δημιουργεί μία στοίβα.
<u>Round 2:</u> Αφού $3 < 6$, το 3 μπαίνει πάνω από το 6.

```
6
3
```

<u>Round 3:</u> Τώρα $5 > 3$, άρα αναγκαστικά δημιουργούμε μία νέα στοίβα.

```
6 5
3
```

<u>Round 4:</u> Τραβάμε 2. Παρατηρείστε ότι η εξέλιξη του παιχνιδιού δεν είναι μοναδική :

6 5 3 2	ή	6 5 3 2

...

Ερώτημα 1.
Input : $d = c_1 c_2 ... c_N$
(GS) GreedyStrategy ($d = c_1 c_2 ... c_N$):

1. Initialization : Δημιούργησε μία στοίβα με την κάρτα c_1.
2. Step : Τοποθετούμε την k-οστή κάρτα c_k πάνω στην μικρότερης αξίας κάρτα που είναι μεγαλύτερη από την c_k. Διαφορετικά (αν δεν υπάρχει τέτοια), ανοίγουμε νέα στοίβα και τοποθετούμε την c_k την κορυφή της.

- Έστω ότι η GS δεν είναι η βέλτιστη στρατηγική. Έστω OPT ο βέλτιστος αλγόριθμος που δημιουργεί λιγότερες στοίβες.
- Ας υποθέσουμε, ότι σε κάποιον γύρο, οι δύο στρατηγικές διαφοροποιούνται. Έστω ότι η άπληστη στρατηγική τοποθετεί το τρέχον φύλλο c στην στοίβα p, ενώ ο 'βέλτιστος' αλγόριθμος την τοποθετεί στην στοίβα q. (c_p, c_q top cards).

$$c < c_p < c_q$$

- Μπορείτε να σχεδιάσετε στρατηγική OPT' που να αποδίδει τουλάχιστον όσο καλά αποδίδει η OPT?
- Strategy Stealing Argument
- Time Complexity : $O(N \log N)$.
- *Παρατήρηση :* Σε κάθε βήμα του αλγορίθμου GS, η ακολουθία των top cards $\{c_{i,|p_i|}\}_{i=1}^{m}$ είναι άυξουσα από αριστερά προς τα δεξιά και κάθε στοίβα είναι προφανώς φθίνουσα.

Ερώτημα 2
Ιδέα A : k-way merge - Time : $O(N \log k)$. Κάνε Merge (η οποία αντιστοιχεί στην 2-way merge) σε ζεύγη πινάκων και άρα, σε κάθε επανάληψη, το πλήθος των πινάκων πέφτει στο μισό.

Ιδέα B : Direct k-way merge - Time : $\Theta(N \log k)$.

Βήματα :

- Έχοντας k στοίβες, δημιουργούμε ένα δέντρο σε $\Theta(k)$.
- Το δέντρο είναι balanced, άρα από τα φύλλα στην ρίζα $\Theta(\log k)$.
- Συνολικά, μεταφέρουμε N στοιχεία (φύλλα) άρα $\Theta(N \log k)$.

Παράδειγμα :
Είσοδος : 2 1 10 5 3 4 9 8 7 6.

$p_1 = \{1, 2\}$, $p_2 = \{3, 5, 10\}$, $p_3 = \{4\}$, $p_4 = \{6, 7, 8, 9\}$

Φτιάχνω balanced δέντρο με $k = 4$ φύλλα. Στο φύλλο leaf$_i$, τοποθετώ το $\min p_i$ και το αφαιρώ από την αντίστοιχη λίστα. Τα φύλλα παίζουν ένα τουρνουά και ανεβάζω στην ρίζα το ελάχιστο όλων. Αν το ελάχιστο προήλθε από την λίστα p_j, στην θέση του leaf$_j$ θέτω το νέο ελάχιστο της λίστας j.

Ερώτημα 3

	3	4	7	8
Παράδειγμα :	2		5	6
	1			

$\triangleright$ Στοίβες = 4

$\triangleright$ $len(LIS)$ = 4 = length($2 \rightarrow 4 \rightarrow 5 \rightarrow 6$) = length($3 \rightarrow 4 \rightarrow 5 \rightarrow 6$) = ...

Αν παίξουμε σύμφωνα με την GS, τότε #piles = $len(LIS)$.

Ερώτημα 4
Αν $c_{i_1}, c_{i_2}, ..., c_{i_L}$ μία μέγιστη αύξουσα υπακολουθία της τράπουλας $c_1, ..., c_N$, τότε σύμφωνα με τους κανόνες του παιχνιδιού και έχοντας τοποθετήσει τις κάρτες $c_{i_1}, c_{i_2}, .., c_{i_k}$, η τοποθέτηση της κάρτας c_{i_k} μας επιβάλλει να ανοίξουμε νέα στοίβα.

Ερώτημα 5
▷ Σε κάθε στάδιο του άπληστου αλγορίθμου, οι top cards αποτελούν μία αύξουσα ακολουθία από αριστερά προς τα δεξιά.

▷ Ισχύει : #piles $\geq$ length of any increasing subsequence
Γιατί? Σε κάθε στοίβα, οι κάρτες σχηματίζουν μία φθίνουσα ακολουθία. Κάθε πιθανή αύξουσα ακολουθία χρησιμοποιεί το πολύ ένα φύλλο από κάθε στοίβα. Η ιδιότητα αυτή ονομάζεται weak duality.

▷ min{#piles} $=$ max{length of any increasing subsequence} και η GS τα υπολογίζει. Γιατί? Κάθε φορά που εισάγεται μία κάρτα σε κάποια στοίβα με $id = p$, τοποθετούμε έναν δείκτη προς την κάρτα που βρίσκεται στην κορυφή της στοίβας με $id = p - 1$ (στην αριστερή της). Ακολουθώντας τα μονοπάτια από αριστερά προς τα δεξιά, σχηματίζουμε αύξουσες ακολουθίες. Αν αρχίσουμε από την δεξιότερη στοίβα, θα πάρουμε ακολουθία με μήκος όσο το πλήθος των στοιβών. Από weak duality, οι δύο ποσότητες είναι optimal. (strong duality).

Ερώτημα 6
▷ $\{1, 2, ..., mn + 1\}$. Εκτελούμε την GS για την τράπουλα με $mn + 1$ φύλλα. Το πλήθος των στοιβών είναι όσο το $len(LIS)$ και η κάθε στοίβα αντιστοιχεί σε μία φθίνουσα ακολουθία.

▷ Δείτε ακόμη το Erdős–Szekeres Theorem.

▷ Ένα παράδειγμα μπορεί να κατασκευαστεί ως εξής ([5,4,3,2,1], [10,9,8,7,6], ...)

Άσκηση 4 - Γρήγορη Επιλογή στο Πεδίο της Μάχης

- **Ερώτημα 1:** Μπορούμε να βρούμε με ακρίβεια τη θέση οποιουδήποτε σωματιδίου οποιαδήποτε χρονική στιγμή σε $O(1)$.
- **Ερώτημα 2:** Αν μας δοθούν 2 σωματίδια διαφορετικού τύπου, μπορούμε με ακρίβεια να βρούμε πότε θα συγκρούονταν (αν δεν γίνονταν οι εξαϋλώσεις) σε $O(1)$.

Ερώτημα 1

- Παρατηρούμε πως η πρώτη σύγκρουση θα πρέπει να γίνει κάποια στιγμή στο διάστημα $\left[\frac{L}{2V_{max}}, \frac{L}{2V_{min}}\right]$.
- Θεωρώντας κάποια ακρίβεια (πχ $a = 10^{-4}$), έχουμε ένα σύνολο από
$$\frac{1}{a}\left(\frac{L}{2V_{min}} - \frac{L}{2V_{max}}\right) + 1 = \Theta\left(L\left(\frac{1}{V_{min}} - \frac{1}{V_{max}}\right)\right)$$
χρονικές στιγμές.

- Κάνουμε δυαδική αναζήτηση σε αυτό το χρονικό διάστημα.
- Αν κάποια στιγμή όλα τα σωματίδια της 1ης κατηγορίας βρίσκονται πιο αριστερά από τα αντίστοιχα της 2ης, κοιτάμε μεταγενέστερες χρονικές στιγμές. Αλλιώς, κοιτάμε προγενέστερες.
- Σε κάθε βήμα, $O(n)$ χρόνος για να βρούμε τις θέσεις των στοιχείων $\implies O\left(n \log \left(L\left(\frac{1}{V_{min}} - \frac{1}{V_{max}}\right)\right)\right)$ συνολικά στη **χειρότερη περίπτωση**.

Ερώτημα 2

- **Απλή Λύση:** Brute Force $\rightarrow$ δοκίμασε όλα τα n^2 ζεύγη (πολυπλοκότητα $O(n^2)$).
- **Καλή Λύση:** Quickselect στα σωματίδια της 1ης κατηγορίας (πολυπλοκότητα $O(n \log n)$).
- **Βέλτιστη Λύση:** Quickselect στα σωματίδια και των 2 κατηγοριών εναλλάξ (πολυπλοκότητα $O(n)$).

Ιδέα Καλής Λύσης (i)

- Επέλεξε τυχαία ένα σωματίδιο της 1ης κατηγορίας.
- Βρες πότε θα συγκρουόταν με καθένα από τα σωματίδια της 2ης κατηγορίας (αν δεν είχαμε τις εξαϋλώσεις) και κράτα εκείνο όπου αντιστοιχεί ο μικρότερος χρόνος σύγκρουσης (έστω j^* το σωματίδιο και t^* ο χρόνος).
- Ισχυρισμός: Δεν υπάρχουν σωματίδια της 2ης κατηγορίας που να βρίσκονται πιο αριστερά από το j^* τη στιγμή t^*.

Ιδέα Καλής Λύσης (ii)

- Πράγματι, αν υπήρχαν τέτοια σωματίδια, τότε το τυχαία επιλεγμένο σωματίδιο θα συγκρουόταν με καθένα από αυτά κάποια στιγμή $t < t^*$ (άτοπο).
- Άρα, τα σωματίδια της 1ης κατηγορίας που βρίσκονται αριστερά του σημείου σύγκρουσης δεν μπορούν να συμμετέχουν στην πρώτη σύγκρουση (αφού κάθε στιγμή $t < t^*$ όλα τα σωματίδια της 2ης κατηγορίας θα βρίσκονται πιο δεξιά).
- Έτσι, μπορούμε τα σωματίδια αυτά να τα αγνοήσουμε.
- Επαναλαμβάνοντας το ίδιο όσες φορές χρειαστεί ώστε να μείνει μόνο ένα σωματίδιο της 1ης κατηγορίας, μπορούμε μετά να βρούμε με ποιο από αυτά της 2ης συγκρούεται πρώτα.

Ανάλυση Πολυπλοκότητας

- Μετά από κάθε βήμα, με σταθερή πιθανότητα θεωρούμε πως μένουν το πολύ τα $\frac{3}{4}$ των σωματιδίων της 1ης κατηγορίας (βλ. ανάλυση Quickselect).

- Άρα, έχουμε $O(\log n)$ επαναλήψεις.
- Σε κάθε επανάληψη, κάνουμε μία γραμμική αναζήτηση στα στοιχεία της 2ης κατηγορίας όπου κάθε σύγκρουση υπολογίζεται σε $O(1)$ χρόνο (άρα $O(n)$ συνολικά).
- Για τα σωματίδια της 1ης κατηγορίας (που σε κάθε επανάληψη είναι $\leq n$), υπολογίζουμε τη στιγμή που συγκρούονται με το j^*. Αν είναι $> t^*$ καταλαβαίνουμε πως είναι αριστερά από το σημείο σύγκρουσης, οπότε ξεσκαρτάρονται.
- Συνολικός Χρόνος: $O(n \log n)$ στη **μέση περίπτωση**.

Ιδέα Βέλτιστης Λύσης

- Ο λόγος που πριν είχαμε πολυπλοκότητα $O(n \log n)$ ήταν επειδή έπρεπε να κάνουμε γραμμική αναζήτηση στα σωματίδια της 2ης κατηγορίας (που ήταν πάντα n).
- Χρησιμοποιούμε την ίδια τεχνική με την προηγούμενη λύση, αλλά την εφαρμόζουμε εναλλάξ στα σωματίδια και των 2 κατηγοριών.
- Μειώνεται παράλληλα το πλήθος των υποψήφιων σωματιδίων και από τις 2 κατηγορίες.
- Τελική Πολυπλοκότητα: $O(n)$ στη **μέση περίπτωση**.

Άσκηση 5 - Ερωτήματα Ανήκειν

- Το (b) αποτελεί φυσική γενίκευση του (a), άρα απαντάμε απ' ευθείας το (b).
- $S \subset \mathbb{N}_+$ τ.ω. $|S| = n$ (κλειδιά).
- Έστω πίνακας A μεγέθους m, αρχικοποιημένος με 0, και $k \geq 1$ συναρτήσεις κατακερματισμού $h_i : \mathbb{N}_+ \to [m]$ τ.ω. $\Pr[h_i(x) = j] = 1/m$ για κάθε $x \in \mathbb{N}_+, j \in [m]$.
- Η αρχικοποίηση της δομής γίνεται ως εξής: Για κάθε κλειδί $x \in S$, υπολογίζουμε το διάνυσμα:

$$x \to (h_1(x), \ldots, h_k(x)) \tag{1}$$

  και υλοποιούμε τη μέθοδο $Insert(x)$: Εισάγουμε 1 στο πίνακα A στις k θέσεις $h_i(x)$.
- $Query(x)$: Για να ελέγξουμε αν το δοθέν κλειδί x ανήκει στο S, ελέγχουμε αν $A[h_1(x)] \wedge \cdots \wedge A[h_k(x)] = 1$. Απαντάμε "ναι" ή "όχι" αναλόγως.
- Η δομή αυτή απαντάει πάντα ορθά αν το κλειδί ανήκει στο σύνολο. Υπάρχει όμως και το ενδεχόμενο να απαντήσει "ναι", ακόμα και αν $x \notin S$ (false positive).
- Για να τεθεί ένα τυχόν bit $i \in [m]$ ίσο με 1, αρκεί ένα από τα n κλειδιά να γίνει map στο bit αυτό. Αυτό συμβαίνει με πιθανότητα:

$$\Pr[A[i] = 1] = 1 - (1 - \frac{1}{m})^{nk} \simeq 1 - e^{-nk/m} \tag{2}$$

  Παρατηρείστε τι συμβαίνει όταν m είτε n αποκλείνουν στο ∞.

- $\Pr[\{A[h_1(x) = 1, \ldots, A[h_k(x) = 1]\}] = \prod_{i=1}^{k} \Pr[A[h_i(x) = 1]] \simeq (1 - e^{-nk/m})^k$.
- Βελτιστοποιούμε την συνάρτηση ως προς k (παρατηρείστε ότι είναι κυρτή), θέτοντας $m = 8n$:

$$\frac{d}{dk}(1 - e^{-k/8})^k = 0 \Rightarrow k \simeq 5.54$$

- Η παραπάνω δομή ονομάζεται Bloom filter.

Άσκηση 6 - Προθέματα Συμβολοσειρών

Κατασκευάζουμε, σε χώρο και χρόνο $O(N)$, ένα trie για τις n συμβολοσειρές εισόδου. Στη συνέχεια, με χρόνο προεπεξεργασίας $O(N)$, αρχικοποιούμε μια δομή για απαντήσεις lowest common ancestor (LCA) ερωτημάτων πάνω στο trie. Για κάθε query, υπολογίζουμε σε χρόνο $O(k)$ το LCA των κόμβων του trie στους οποίους καταλήγουν οι συμβολοσειρές $s_1, \ldots, s_k$.