Αλγόριθμοι και Πολυπλοκότητα – τελικές εξετάσεις χειμερινού εξαμήνου 2023-2024

Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Πανεπιστημίου Ιωαννίνων

29/1/2024 (διάρκεια εξέτασης 3 ώρες)

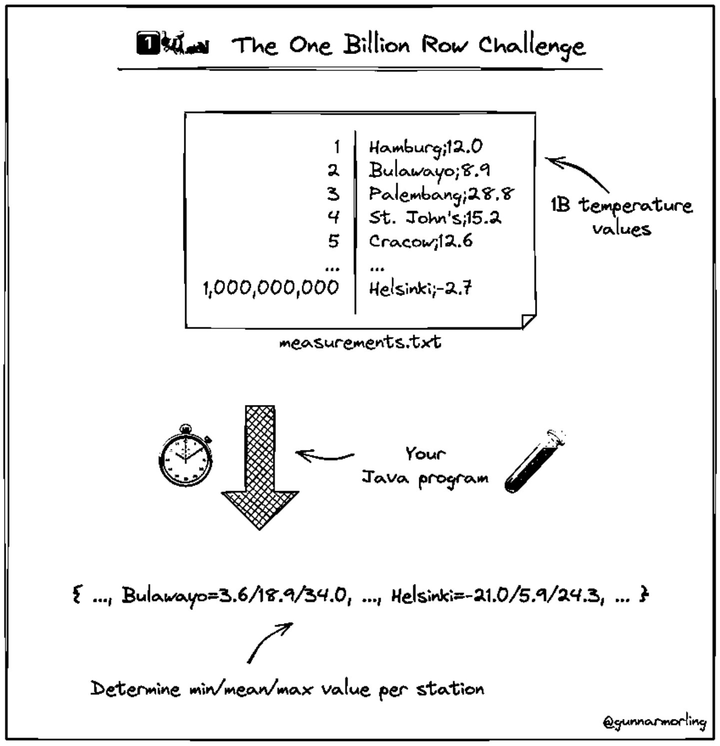
Θέμα 1 [1: 1 μονάδα, 2α+2β: 1 μονάδα]

1. Σχεδιάστε πρόχειρα σε ένα σύστημα αξόνων τις συναρτήσεις: n, 2^n, logn. Γραμμοσκιάστε την περιοχή στην οποία αντιστοιχεί το O(logn).

2. Τον Ιανουάριο του 2024, απέκτησε δημοσιότητα το 1BRC (1 Billion Row Challenge) όπου ζητείται μια αποδοτική Java υλοποίηση ενός αλγορίθμου για τη εύρεση της ελάχιστης, μέσης και μέγιστης τιμής ανά τοποθεσία για 1 δισεκατομμύριο παρατηρήσεις θερμοκρασιών (12GB δεδομένων).

α) Περιγράψτε έναν αποδοτικό τρόπο επίλυσης του προβλήματος.

β) Η ταχύτερη λύση που έχει κατατεθεί υπολογίζει τα αποτελέσματα σε λιγότερο από 2 δευτερόλεπτα χρησιμοποιώντας 8 από τους 32 πυρήνες ενός σύγχρονου υπολογιστή (AMD EPYC™ 7502P) με 128GB μνήμης RAM. Ποια εκτιμάτε ότι είναι η πολυπλοκότητα του αλγορίθμου για τη συγκεκριμένη λύση;



Θέμα 2 [1α+1β: 1μονάδα, 2α+2β: 2 μονάδες]

1. Σύμφωνα με το Master θεώρημα ισχύει ότι:

$$T(n) = \begin{cases} c & \text{if } n < d \\ aT(n/b) + f(n) & \text{if } n \geq d \end{cases}$$

1. if $f(n)$ is $O(n^{\log_b a - \varepsilon})$, then $T(n)$ is $\Theta(n^{\log_b a})$
2. if $f(n)$ is $\Theta(n^{\log_b a} \log^k n)$, then $T(n)$ is $\Theta(n^{\log_b a} \log^{k+1} n)$
3. if $f(n)$ is $\Omega(n^{\log_b a + \varepsilon})$, then $T(n)$ is $\Theta(f(n))$, provided $af(n/b) \leq \delta f(n)$ for some $\delta < 1$.

α) Σε τι είδους προβλήματα εφαρμόζεται το Master θεώρημα και τι αναπαριστούν τα a, b και f(n) στον παραπάνω τύπο;

β) Ποια είναι η πολυπλοκότητα της T(n)=T(n/2) + 1; Περιγράψτε το πως φτάσατε στο συμπέρασμα;

2. Θεωρείστε ότι η ακόλουθη λίστα γειτονικότητας αναπαριστά έναν κατευθυνόμενο γράφημα.

```
graph = {
    "A": ["B", "C"],
    "B": ["C", "D"],
    "C": ["D"],
    "D": ["E"],
    "E": [] }
```

α) Σχεδιάστε το γράφημα. Ποια θα είναι τα περιεχόμενα ενός πίνακα γειτονικότητας για το ίδιο γράφημα;

β) Γράψτε κώδικα που να υλοποιεί τη συνάρτηση areAdjacent(v,w) που εξετάζει αν δύο κορυφές v και w είναι γειτονικές ή όχι, επιστρέφοντας True ή False αντίστοιχα (χρησιμοποιώντας τη λίστα γειτονικότητας). Καλέστε τη συνάρτηση για κάθε ζεύγος κορυφών.

Θέμα 3 [1α+1β=1 μονάδα, 2: 1μονάδα]

1. Ποια είναι η διαφορά ανάμεσα στο πρόβλημα 0/1 σακιδίου (0/1 knapsack) και στο πρόβλημα κλασματικού σακιδίου (fractional knapsack) α) σε σχέση με το πρόβλημα που επιλύεται και β) σε σχέση με τη μέθοδο επίτευξης της βέλτιστης λύσης;

2. Δίνεται ότι η αναδρομική εξίσωση για την επίτευξη της βέλτιστης λύσης στην περίπτωση του 0/1 σακιδίου είναι:

B[k,w]=B[k-1,w] αν $w_k > w$
B[k,w]=max(B[k-1,w], B[k-1,w-w_k]+b_k) αν $w_k <= w$

Έστω η ακόλουθη λίστα 4 αντικειμένων με βάρη και τιμές και ένα σακίδιο με χωρητικότητα W=4.

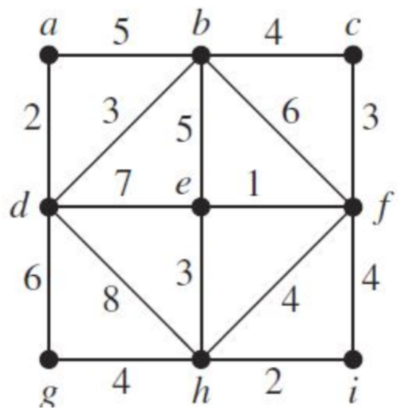
Αντικείμενο	Βάρος	Αξία
1	3	1
2	1	2
3	4	3
4	2	4

Συμπληρώστε τον ακόλουθο πίνακα B για την εύρεση της βέλτιστης λύσης, κυκλώνοντας τα κελιά που εντοπίζουν τη βέλτιστη λύση:

B	0	1	2	3	4
0					
1					
2					
3					
4					

Θέμα 4 [α: 2 μονάδες, β: 1 μονάδα]

Δίνεται το ακόλουθο γράφημα:



α) Εφαρμόστε τον αλγόριθμο του Dijkstra για την εύρεση των συντομότερων διαδρομών με αφετηρία την κορυφή a. Καταγράψτε για κάθε κορυφή v όλες τις τιμές που σταδιακά λαμβάνει η ετικέτα D[v] (δλδ η απόσταση της κορυφής v από την κορυφή αφετηρίας. Ποια είναι η χρονική πολυπλοκότητα του αλγορίθμου του Dijkstra για την εύρεση των συντομότερων διαδρομών σε έναν γράφο; Αναφέρατε πλεονεκτήματα και μειονεκτήματα του αλγορίθμου.

β) Εφαρμόστε τον αλγόριθμο του Prim για τον εντοπισμό του ελάχιστου συνεκτικού δέντρου χρησιμοποιώντας ως αφετηρία την κορυφή a. Σχεδιάστε το ελάχιστο συνεκτικό δέντρο που προκύπτει και υπολογίστε το συνολικό βάρος των ακμών του. Καταγράψτε τις ακμές που ορίζουν το ελάχιστο συνεκτικό δέντρο με τη σειρά με την οποία ο αλγόριθμος του Prim τις προσαρτά σε αυτό. Αναφέρατε έναν άλλο αλγόριθμο που επιλύει το ίδιο πρόβλημα.