Αλγόριθμοι και Πολυπλοκότητα

$2^η$ Εργασία

Κωστόπουλος Κωνσταντίνος

03117043

Άσκηση 1

Για κάθε σημείο μπορούμε να υπολογίσουμε ένα διάστημα πάνω στην ευθεία, εντός του οποίου αν τοποθετηθεί κέντρο δίσκου, τότε αυτός θα καλύπτει το σημείο. Αρκεί να υπολογίσουμε τα δύο ακραία σημεία του διαστήματος για κάθε σημείο (x_i, y_i)

$$s_i = x_i - \sqrt{r^2 - y_i^2}, \quad f_i = x_i + \sqrt{r^2 - y_i^2}$$

Το *Greedy* κριτήριο είναι η τοποθέτηση δίσκων κάθε φορά στο ελάχιστο f_i που έχει απομείνει. Ο αλγόριθμος είναι ο εξής

- Ταξινόμηση σημείων ως προς f_i
- Τοποθέτηση δίσκου στο ελάχιστο f_i
- Διαγραφή των επόμενων σημείων με $s_j \leq f_i$

Η πολυπλοκότητα είναι $O(n \log n)$, αφού η τοποθέτηση δίσκων γίνεται με ένα γραμμικό πέρασμα, αφού τελειώσει η ταξινόμηση.

Απόδειξη ορθότητας κριτηρίου:
Ο επόμενος δίσκος που προσθέτουμε κάθε φορά, δεν μπορεί να προστεθεί πιο δεξιά από το τέλος του διαστήματος που τελειώνει πρώτο, αλλιώς δεν θα καλυφθεί το αντίστοιχο σημείο. Αν προστεθεί πιο αριστερά, καλύπτει το πολύ όσα θα κάλυπτε αν είχε προστεθεί στο τέλος του διαστήματος, εφόσον είναι το διάστημα που τελειώνει πρώτο.

Άσκηση 2

(α)

1. Η στοίβαξη με βάση το βάρος είναι λανθασμένο κριτήριο.

   Αντιπαράδειγμα:
   Έστω ότι έχουμε δύο αντικείμενα $(3,1),(2,3)$. Τότε υπάρχει ασφαλής στοίβαξη και είναι $(2,3) \rightarrow (3,1)$. Όμως λόγω του κριτηρίου θα δημιουργηθεί η $(3,1) \rightarrow (2,3)$ που δεν είναι ασφαλής.

2. Η στοίβαξη με βάση την αντοχή είναι λανθασμένο κριτήριο.

   Αντιπαράδειγμα:
   Έστω ότι έχουμε δύο αντικείμενα $(3,1),(1,2)$. Τότε υπάρχει ασφαλής στοίβαξη και είναι η $(3,1) \rightarrow (1,2)$. Όμως λόγω του κριτηρίου θα δημιουργηθεί η $(1,2) \rightarrow (3,1)$ που δεν είναι ασφαλής.

3. Η στοίβαξη με βάση το άθροισμα βάρους και αντοχής είναι σωστό κριτήριο.

   Απόδειξη:
   Αρκεί να δειχθεί ότι με αλλαγή της σειράς των αντικειμένων σε ασφαλή διάταξη, μπορεί να δημιουργηθεί νέα ασφαλής διάταξη σύμφωνα με το κριτήριο. Έστω μια ασφαλής διάταξη

   $$(w_1,d_1) \rightarrow \cdots \rightarrow (w_j,d_j) \rightarrow \cdots \rightarrow (w_n,d_n) \qquad j = argmax_{1\leq i\leq n}(w_i+d_i)$$

   Έστω

   $$W_1 = \sum_{i=2}^{j-1}(w_i+d_i), \quad W_2 = \sum_{i=j+1}^{n}(w_i+d_i)$$

   Εφόσον είναι ασφαλής θα ισχύει

   $$\begin{aligned} & W_1 + w_j + W_2 \leq d_1 \Leftrightarrow \\ & w_j \leq d_1 - W_1 - W_2 \end{aligned}$$

   Λόγω της επιλογής του j και της παραπάνω σχέσης

   $$\begin{aligned} & w_1 + d_1 \leq w_j + d_j && \Leftrightarrow \\ & w_1 + d_1 \leq d_1 - W_1 - W_2 + d_j && \Leftrightarrow \\ & W_1 + w_1 + W_2 \leq d_j \end{aligned}$$

   Άρα μπορούμε απλά να μετακινήσουμε το αντικείμενο j στην πρώτη θέση χωρίς να γίνει άλλη αλλαγή και να έχουμε ασφαλή στοίβαξη. Μπορούμε να επαναλάβουμε την ίδια διαδικασία n φορές, αφαιρώντας κάθε φορά το στοιχείο με το μέγιστο άθροισμα και τοποθετώντας το στην επόμενη θέση μιας νέας στοίβας. Σε περίπτωση που έχουν ίδιο μέγιστο άθροισμα πολλά στοιχεία αρκεί να επιλέγουμε με βάση τη μεγαλύτερη αντοχή ώστε να είναι μοναδική η στοίβαξη που παράγει το κριτήριο.

(β)

Στην περίπτωση που δεν υπάρχει ασφαλής στοίβαξη για όλα τα στοιχεία, θα πρέπει για κάθε στοιχείο να επιλέξω αν θα προστεθεί στην τελική στοίβα ή όχι. Αν προστεθεί, τότε η νέα ελάχιστη αντοχή θα προκύψει είτε από την αφαίρεση της προηγούμενης και της τωρινής είτε από την τωρινή. Έστω $P(i,d)$ το μεγαλύτερο κέρδος που μπορεί να προκύψει από τα πρώτα i στοιχεία, αν το επιπλέον βάρος μπορεί να είναι συνολικά το πολύ ίσο με d. Άρα

$$P(i,d) = \begin{cases} \max\{p_i + P(i-1, \min\{d_i, d-d_i\}), P(i-1,d)\} & \text{αν } d_i \leq d \\ P(i-1,d) & \text{αν } 0 < d < d_i \\ 0 & \text{αν } d = 0 \text{ ή } i = 0 \end{cases}$$

Η πολυπλοκότητα προκύπτει $\Theta(n \cdot d_{max})$ όπου d_{max} η μέγιστη αντοχή αντικειμένου, επειδή υπάρχουν $n \cdot d_{max}$ καταστάσεις και το κάθε βήμα έχει σταθερό χρόνο.

Άσκηση 3

Πρέπει κάθε φορά να χωρίσουμε το κυρτό πολύγωνο σε δύο κυρτά πολύγωνα, συνδέοντας δύο κορυφές με τον βέλτιστο τρόπο.

Έστω $C(i,j)$ το ελάχιστο κόστος τριγωνοποίησης του κυρτού πολυγώνου με κορυφές αυτές που συναντάμε προχωρώντας με αντίθετη φορά του ρολογιού από την v_i ως την v_j. Προφανώς, μπορεί $i \geq j$.

Το παραπάνω σύνολο κορυφών μπορεί να ονομαστεί $t(i,j)$. Τότε $t(i,i)$ είναι μία πλήρης κυκλική διαδρομή.

Άρα

$$C(i,j) = \begin{cases} \min_{k \in t(i,j)} \{||v_i v_k|| + C(i,k) + C(k,i)\} & \text{αν } |t(i,j)| \geq 2 \\ 0 & \text{αν } |t(i,j)| \leq 1 \end{cases}$$

Η πολυπλοκότητα του παραπάνω αλγορίθμου είναι $O(n^3)$ επειδή υπάρχουν n^2 καταστάσεις και κάθε φορά κάνουμε ένα γραμμικό πέρασμα για να υπολογίσουμε μια νέα κατάσταση.

Η λύση του προβλήματος μπορεί να βρεθεί με διάσχιση του πίνακα στα στοιχεία της διαγωνίου, καθώς πάντα θα υπάρχουν δύο κορυφές από τις οποίες δεν ξεκινούν πλευρές τριγώνων. Επίσης, κάθε χορδή ανήκει σε δύο τρίγωνα άρα πρέπει να υπολογιστεί δύο φορές. Αφού υπολογίσουμε το ελάχιστο στοιχείο της διαγωνίου, έστω $C(x,x)$ και την περίμετρο του πολυγώνου Π, η το αποτέλεσμα θα είναι

$$2 \cdot C(x,x) + \Pi$$

Άσκηση 4

(α)

Αρκεί να βρούμε τον βέλτιστο τρόπο να τοποθετήσουμε στέγαστρο από καποιον σταθμό x_i μέχρι τον x_j και να λύσουμε το πρόβλημα για τους σταθμούς μετά τον x_j.

Έστω $S(i)$ το ελάχιστο κόστος για να καλυφθούν όλοι οι σταθμοί από τον x_i μέχρι το τέλος.

Άρα

$$S(i) = \begin{cases} \min_{i \leq j \leq n} \{S(j+1) + cost(x_j, x_i)\} & \text{αν } i \leq n \\ 0 & \text{αν } i = n+1 \end{cases}$$

Το πλήθος καταστάσεων είναι n και η κάθε κατάσταση υπολογίζεται σε γραμμικό χρόνο. Άρα η πολυπλοκότητα είναι $O(n^2)$.

(β)

Μπορούμε να αναπαραστήσουμε την κυρτή κάλυψη ως μια λίστα με κόμβους που περιέχουν την ευθεία που είναι ελάχιστη σε κάποιο διάστημα και το σημείο τομής της με την επόμενη ευθεία της κυρτής κάλυψης.

Για να φτιάξουμε αυτή τη λίστα μπορούμε να προσθέτουμε διαδοχικά ευθείες στο τέλος της λίστας και να βρίσκουμε τη θέση στην οποία ανήκουν.

- Βρίσκω σημείο τομής αν υπάρχει, έστω (x, y), ανάμεσα στην ευθεία μας και την τελευταία ευθεία της λίστας.
- Έστω ότι αυτή η ευθεία είχε σημείο τομής με την προηγούμενη το (x', y').
- Αν $x' < x$ τότε η νέα ευθεία μπορεί να προστεθεί στην τελευταία θέση, αφού θα ξεκινήσει να είναι η ελάχιστη ευθεία μετά την ήδη υπάρχουσα.
- Αν $x < x'$ τότε αφαιρούμε την τελευταία ευθεία της λίστας και βάζουμε τη δικιά μας. Επαναλαμβάνουμε μέχρι να μην μπορεί να αντικαταστήσει άλλη ευθεία της λίστας.
- Αν δεν υπάρχει σημείο τομής, δηλαδή είναι παράλληλες, τότε επιλέγουμε αυτή με το μικρότερο b και συνεχίζουμε μέχρι να μην μπορεί να αντικαταστήσει άλλη ευθεία.

Αυτή η διαδικασία γίνεται σε γραμμικό χρόνο, αφού κάθε φορά που διασχίζουμε τη λίστα αφαιρούμε όσα στοιχεία περάσαμε.

Για να βρούμε την ελάχιστη ευθεία για κάθε σημείο μπορούμε να κάνουμε γραμμικό πέρασμα στη λίστα συνεχίζοντας κάθε φορά από εκεί που σταματήσαμε. Άρα η πολυπλοκότητα είναι $\Theta(n + k)$.

Σε σχέση με το πρώτο ερώτημα μπορούμε να ξαναγράψουμε την αναδρομική σχέση ως εξής

$$\begin{aligned} S(i) = \min_{i \le j \le n} \{S(j+1) + cost(x_j, x_i)\} & = \\ = \min_{i \le j \le n} \{S(j+1) + (x_j - x_i)^2 + c\} & = \\ = \min_{i \le j \le n} \{S(j+1) + x_j^2 - 2x_j \cdot x_i\} + x_i^2 + c & \end{aligned}$$

Αν

$$\begin{aligned} c_i &= x_i^2 + c \\ a_j &= -2x_j \\ b_j &= x_j^2 + S(j-1) \end{aligned}$$

Τότε

$$S(i) = \min_{i \le j \le n} \{a_j \cdot x_i + b_j\} + c_i$$

Για τον υπολογισμό της κάθε κατάστασης, μπορούμε να προσθέτουμε για κάθε i την ευθεία (a_i, b_i) στη λίστα όπως περιγράφηκε παραπάνω. Παράλληλα μπορούμε να μετακινούμε κάποιον δείκτη που αντιστοιχεί στο x_i για να βρούμε σε ποιά ευθεία ελαχιστοποιείται η τιμή. Κάθε φορά ο δείκτης μπορεί να συνεχίζει είτε από τη θέση που είχε μείνει είτε από την τελευταία θέση αν διαγράφηκαν πολλά στοιχεία της λίστας. Είναι προφανές ότι όπως στα παραπάνω τα a_i είναι αρνητικά και σε φθίνουσα σειρά, ενώ τα x_i σε αύξουσα σειρά.Τελικά η πολυπλοκότητα θα είναι $\Theta(n)$.

Άσκηση 5

(α)

Κάθε κόμβο είτε τον προσθέτω στο σύνολο K είτε όχι. Για κάθε κόμβο μπορώ να κρατάω την απόσταση του απο τον κοντινότερο κόμβο που έχει προστεθεί στο σύνολο. Αν τον προσθέσω στο σύνολο πρέπει να μηδενίσω την απόσταση αυτή. Σε κάθε περίπτωση συνεχίζω αναδρομικά στα υποδέντρα μέχρι να συμπληρώσω τους k κόμβους με ελάχιστο μέγιστο κόστος.

Έστω $P(x, c, k)$ το ελάχιστο μέγιστο κόστος κόμβου που ανήκει στο δέντρο με ρίζα x, αν μένουν ακόμα k κόμβοι να προστεθούν στο σύνολο και ο κοντινότερος πρόγονος του x στο K έχει απόσταση c.

Άρα

$$P(x,c,k) = \begin{cases} \min\{P(x,0,k), \quad \min_i\{\max\{P(left,c+1,i), P(right,c+1,k-i)\}\}\} & \text{αν } c > 0 \\ \min_i\{\max\{P(left,c+1,i), P(right,c+1,k-i)\}\} & \text{αν } c = 0 \\ c & \text{αν } x = null \end{cases}$$

Άρα έχουμε $n^2 \cdot k$ καταστάσεις και κάθε κατάσταση υπολογίζεται σε γραμμικό χρόνο. Άρα πολυπλοκότητα $\Theta(n^2 \cdot k^2)$.

(β)

Μπορούμε να χωρίσουμε το δέντρο σε επίπεδα, και να βρούμε πόσοι κόμβοι περιέχονται σε κάθε επίπεδο σε γραμμικό χρόνο. Για κάθε επίπεδο είτε προσθέτω όλους τους κόμβους του στο K είτε κανέναν. Κάθε φορά διατηρώ την απόσταση των διαδοχικών επιπέδων που έχουν προστεθεί στο K το πολύ ιση με z.

Για την παραμετροποίηση του προβλήματος, έστω $C(i, d)$ το ελάχιστο πλήθος κόμβων που πρέπει να προστεθούν στο K από το επίπεδο i και κάτω, αν μπορούμε να προχωρήσουμε το πολύ d επίπεδα χωρίς να προσθέσουμε κάποιο επίπεδο στο K.

Άρα

$$C(i,d) = \begin{cases} \min\{C(i+1,d-1), C(i+1,z)\} & \text{αν } d > 0 \\ C(i+1,z) & \text{αν } d = 0 \end{cases}$$

Η πολυπλοκότητα θα είναι $\Theta(n \cdot z)$.

Αυτό μπορεί να χρησιμοποιηθεί στο ερώτημα (α), κάνοντας *binary search* στην τιμή του κόστους ως εξής.

Στο βήμα που θα δοκιμάσουμε κόστος x, θα βρούμε μέγεθος του K ίσο με s σε χρόνο $O(n \cdot x)$.

- Αν $s > k$ τότε απαιτούνται περισσότεροι κόμβοι για το συγκεκριμένο κόστος οπότε συνεχίζουμε στο δεξί μισό.
- Αν $s \leq k$ τότε δοκιμάζω να χαμηλώσω περισσότερο το κόστος πηγαίνοντας στο αριστερό μισό.

Συνολικά η πολυπλοκότητα θα είναι $O(n^2 \cdot \log n)$, επειδή το μέγιστο κόστος είναι ίσο με το πλήθος των κόμβων στη χειρότερη περίπτωση.