Online Διαγωνισμοί HelleniCO

Νοέμβριος 2012

Εγχειρίδιο Λύσεων

Περιεχόμενα

Μπορείτε να συζητήσετε τις απορίες σας πάνω στις λύσεις και τα θέματα στη διεύθυνση
`http://www.pdpforum.eu.org/forum/viewtopic.php?f=3&t=2258`

1 Ψηφιδωτό (mosaic)

Δημιουργός: Κυριάκος Αξιώτης

Εκφώνηση: `http://www.hellenico.gr/contest/task.php?name=mosaic`

Αρχικά διαβάζουμε το input σε ένα δισδιάστατο πίνακα χαρακτήρων και ψάχνουμε να βρούμε όλους τους χαρακτήρες 'x'. Για κάθε τέτοια θέση στον πίνακα, έστω $A[i][j]$, θα πρέπει να ελέγξουμε τις γειτονικές θέσεις και να μετρήσουμε πόσες γαλάζιες, πράσινες και αλλοιωμένες ψηφίδες υπάρχουν. Για να το κάνουμε αυτό αρκεί να ελέγξουμε τις εξής θέσεις: $A[i][j+1]$, $A[i][j-1]$, $A[i+1][j]$, $A[i+1][j+1]$, $A[i+1][j-1]$, $A[i-1][j]$, $A[i-1][j-1]$, $A[i-1][j+1]$. Στο τέλος της μέτρησης και αν η συνθήκη που θέλουμε για τα πλήθη των ψηφίδων ικανοποιείται, αυξάνουμε το πλήθος των ψηφίδων που βρήκαμε και μπορούμε να μαντέψουμε το χρώμα τους.

```
1 galazies=prasines=alloiwmenes=0;
2 for (dx=-1;dx<=1;++dx) for (dy=-1;dy<=1;++dy) if (dx!=0 || dy!=0) {
3   if (color[x+dx][y+dy]=='g) ++galazies;
4   else if (color[x+dx][y+dy']=='p) ++prasines;
5   else if (color[x+dx][y+dy']=='x) ++alloiwmenes;
6 }
```

Χρονική πολυπλοκότητα: $O(NM)$

2 Jigsaw (jigsaw)

Δημιουργός: Βαγγέλης Κηπουρίδης

Εκφώνηση: `http://www.hellenico.gr/contest/task.php?name=jigsaw`

Στην ουσία έχουμε δύο επιλογές: Ή θα πάμε κατευθείαν στο δωμάτιο που πρέπει να φτάσει ο Μίλτος, ή θα ανοίξουμε πρώτα την πυρασφάλεια και από εκεί θα κατευθυνθούμε στον τελικό προορισμό μας χρησιμοποιώντας και τους διαδρόμους που ενεργοποίησε η πυρασφάλεια. Θεωρούμε τα δωμάτια σαν τους κόμβους (κορυφές) ενός γράφου και τους διαδρόμους μεταξύ τους ως τις ακμές του γράφου αυτού. Η τελική απάντηση του προβλήματος βλέπουμε εύκολα ότι είναι η μικρότερη από δύο τιμές:

- Την ελάχιστη απόσταση από το Α στο Β
- Την ελάχιστη απόσταση από το Α στο Γ + την ελάχιστη απόσταση από το Γ στο Β, χρησιμοποιώντας τους επιπλέον διαδρόμους

Αυτές τις τιμές, εφόσον τα μήκη των ακμών είναι μοναδιαία, μπορούμε να τις υπολογίσουμε κάνοντας 2 φορές **αναζήτηση κατά πλάτος**[1]: Μία φορά ξεκινώντας από το στον αρχικό γράφο και μία φορά ξεκινώντας από το στον ανανεωμένο γράφο. Χρονική πολυπλοκότητα: $O(N + M)$

[1] `http://en.wikipedia.org/wiki/Breadth-first_search`

3 Dancing with the starfish (dance)

Δημιουργός: Γιάννης Χατζημίχος

Εκφώνηση: `http://www.hellenico.gr/contest/task.php?name=dance`

Θεωρούμε $N = L + R$.

3.1 Απλή προσομοίωση

Η πιο απλή λύση είναι προφανώς να προσομοιώσουμε τη διαδικασία βήμα προς βήμα (δευτερόλεπτο προς δευτερόλεπτο), λαμβάνοντας τα επιθυμητά αποτελέσματα. Θα αποδειχθεί αργότερα ότι η όλη διαδικασία θα διαρκέσει το πολύ $-$ δευτερόλεπτα, σε κάθε ένα από τα οποία κινούμε όλους τους N χορευτές. Αυτή η μέθοδος έχει χρονική πολυπλοκότητα $O(N(B-A))$ και φυσικά δεν είναι αρκετά γρήγορη.

3.2 Βελτιωμένη προσομοίωση

***Παρατήρηση* 1.** *Θα γίνουν συνολικά το πολύ $O(N^2)$ συγκρούσεις μεταξύ χορευτών (σύγκρουση θεωρούμε και την έξοδο από την πλατφόρμα).*

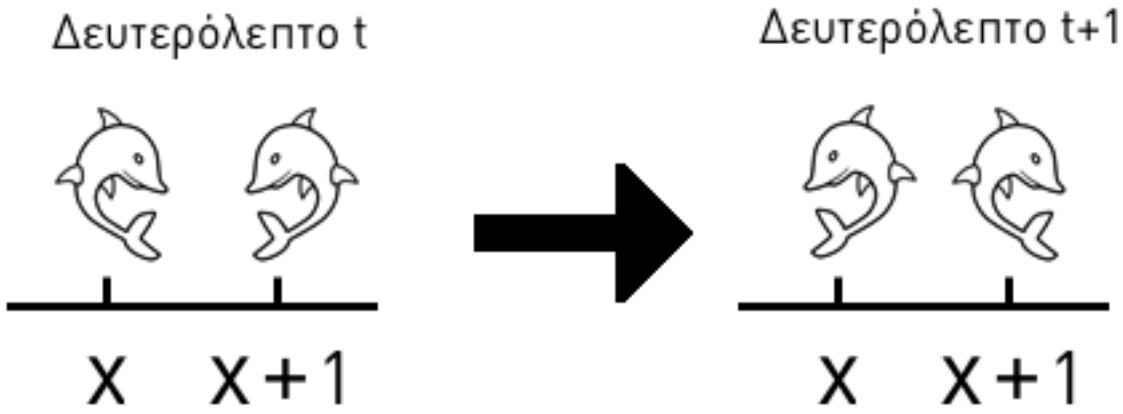
Απόδειξη. Οι ακριανοί χορευτές θα συγκρουστούν το πολύ 2 φορές ο καθένας, οι αμέσως επόμενοι το πολύ 4 φορές, οι αμέσως επόμενοι 6 φορές και ούτω καθεξής. Θα γίνουν δηλαδή το πολύ $2(2+4+6+...+(2\frac{N}{2})) = 4(1+2+3+...+\frac{N}{2}) = 4\frac{N}{4}(\frac{N}{2}+1) = O(N^2)$ συγκρούσεις. □

Μπορούμε λοιπόν να μην κάνουμε προσομοίωση σε κάθε δευτερόλεπτο, αλλά σε κάθε σύγκρουση (όσο δεν γίνονται συγκρούσεις δεν συμβαίνει τίποτα το σημαντικό). Άρα έστω ότι η επόμενη σύγκρουση θα γίνει σε x δευτερόλεπτα από τώρα. Κινούμε όλους τους χορευτές κατά x μέτρα στην κατεύθυνση που κοιτάζουν, αλλάζουμε κατεύθυνση στους δύο χορευτές που μόλις συγκρούστηκαν και βρίσκουμε ποια θα είναι η επόμενη σύγκρουση, συνεχίζοντας την ίδια διαδικασία. Για να βρούμε πότε θα γίνει η επόμενη σύγκρουση αρκεί να εντοπίσουμε το ζευγάρι των διαδοχικών χορευτών από τους οποίους ο αριστερός κινείται δεξιά και ο δεξιός κινείται αριστερά. Χρειαζόμαστε $O(N)$ χρόνο για να υπολογίσουμε όλα τα παραπάνω μετά από κάθε σύγκρουση και μιας και το πλήθος των συγκρούσεων είναι $O(N^2)$, η χρονική πολυπλοκότητα του αλγορίθμου είναι $O(N^3)$.

3.3 Βέλτιστη λύση

***Παρατήρηση* 2.** *Κάθε φορά που συγκρούονται δύο χορευτές, μπορούμε να θεωρήσουμε ότι διαπερνούν ο ένας τον άλλο και συνεχίζουν την πορεία τους.*

Απόδειξη. Αν υπάρχουν 2 χορευτές σε διαδοχικές θέσεις x και $x+1$ που κοιτούν ο ένας τον άλλον, στο επόμενο δευτερόλεπτο θα υπάρχουν χορευτές στις ίδιες θέσεις που βρίσκονται «πλάτη με πλάτη», δηλαδή ακριβώς σαν να πέρασε ο ένας μέσα από τον άλλο. □



Άρα λοιπόν αν βρούμε για κάθε δελφίνι την αρχική του απόσταση από την άκρη της πλατφόρμας που κοιτάει, έχουμε υπολογίσει το χρόνο εξόδου ενός δελφινιού (όχι απαραίτητα του συγκεκριμένου) από την πλατφόρμα. Με αυτό τον τρόπο μπορούμε να υπολογίσουμε όλους τους χρόνους εξόδου από την πλατφόρμα, χωρίς όμως να ξέρουμε σε ποιο δελφίνι αντιστοιχεί ο κάθε χρόνος. Προκειμένου να κάνουμε την αντιστοίχηση, βασιζόμαστε στην παρακάτω παρατήρηση.

***Παρατήρηση* 3.** *Η σχετική θέση των δελφινιών δεν αλλάζει ποτέ. Έτσι, αν κάποιο δελφίνι βρίσκεται αρχικά αριστερά από κάποιο άλλο, δεν θα βρεθεί ποτέ δεξιά από αυτό.*

Αρχικά, ταξινομούμε τους χρόνους εξόδου και τους προσπελαύνουμε σε αύξουσα σειρά. Ακόμη, χρησιμοποιούμε μια μεταβλητή $left$ που αρχικά δείχνει στον αριστερότερο χορευτή και μια μεταβλητή $right$ που δείχνει στον δεξιότερο. Αν η i-οστη έξοδος έγινε από το αριστερό άκρο της πλατφόρμας, τότε αντιστοιχούμε τον χρόνο της με το δελφίνι που δείχνει η μεταβλητή $left$ και στη συνέχεια αυξάνουμε την τιμή της κατά 1 ωστέ να δείχνει στο επόμενο δελφίνι. Όμοια πράττουμε για τις εξόδους που γίνονται από το δεξιό άκρο της πλατφόρμας. Ο αλγόριθμος είναι $O(N)$ και, συμπεριλαμβανομένης της αρχικής ταξινόμησης των δελφινιών, η συνολική χρονική πολυπλοκότητα είναι $O(N \log N)$.