Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Αλγόριθμοι και Πολυπλοκότητα
Διδάσκοντες: Σ. Ζάχος, Δ. Φωτάκης
2η Σειρά Γραπτών Ασκήσεων - Σχέδιο Λύσεων

Άσκηση 1: Ασυμπτωτική Εκτίμηση

Η σειρά κατάταξης για $x > 1$ είναι η ακόλουθη:

$$\frac{\log n}{n^x}, \log n^{x+3}, \log^{x-1} n, \left(\frac{\log n}{\log\log n}\right)^{\frac{\log n}{\log\log n}}, \frac{n}{\log n}, 10n$$

$$n\log n, \binom{n}{x}, \frac{n^{x+1}}{\log^{x+1} n}, 3n^{x+2}, n^{\log\log n}, n2^n, e^n, \sqrt{n!}, n!, 2^{2^n}$$

Για την κατάταξη μπορούμε να χρησιμοποιήσουμε ότι

$$\text{Αν } \lim_{n\to+\infty}\frac{f(n)}{g(n)} = L \text{ με } 0 < L \leq \infty \text{ τότε } f(n) = \Omega(g(n)). \text{ Αν } 0 < L < \infty \text{ τότε } f(n) = \Theta(g(n)).$$

Ακόμα χρήσιμες είναι οι ακόλουθες σχέσεις:

$$\binom{n}{x} = \frac{n!}{x!(n-x)!} = \frac{n(n-1)\dots(n-x+1)}{x!} = \Theta(n^x)$$
$$n! = \Theta\left(\sqrt{2\pi n}\left(\frac{n}{e}\right)^n\right) \qquad \text{(Stirling's approximation)}$$
$$n^{\log\log n} = e^{\log n \log\log n} \qquad \text{(αν θεωρήσουμε φυσικούς λογάριθμους)}$$

$$\left(\frac{\log n}{\log\log n}\right)^{\frac{\log n}{\log\log n}} = e^{\frac{\log n}{\log\log n}(\log\log n - \log\log\log n)} = e^{\log n(1-\frac{\log\log\log n}{\log\log n})}$$

Έτσι έχουμε ότι

$$\frac{n}{\log n} = e^{\log n - \log\log n} >_{\text{ασυμπτωτικά}} e^{\log n(1-\frac{\log\log\log n}{\log\log n})} = \left(\frac{\log n}{\log\log n}\right)^{\frac{\log n}{\log\log n}}$$

και συνεπώς $\left(\frac{\log n}{\log\log n}\right)^{\frac{\log n}{\log\log n}} = o(\frac{n}{\log n})$. Σημειώνουμε ότι για κάθε σταθερά $\varepsilon > 0$, $\left(\frac{\log n}{\log\log n}\right)^{\frac{(1+\varepsilon)\log n}{\log\log n}} = \omega(n)$, δηλαδή αν πολλαπλασιάσουμε τον εκθέτη με οποιαδήποτε σταθερά μεγαλύτερη της μονάδας, η τάξη μεγέθους της συνάρτησης αλλάζει από υπο-γραμμική σε υπερ-γραμμική.

Σημείωση: Για $x = 1$ η κατάταξη αλλάζει ελαφρώς. Πώς;

Άσκηση 2: Επίλυση Αναδρομικών Σχέσεων

(α)

$$\begin{aligned} T(n) &= 2 \cdot T(n/3) + 29n = 2(2 \cdot T(n/3^2) + 29n/3) + 29n \\ &= 2(2(2 \cdot T(n/3^3) + 29n/3^2) + 29n/3) + 29n = \dots \\ &= 2^{i+1} \cdot T(n/3^{i+1}) + 29n \sum_{k=0}^{i} (2/3)^k \\ &= 2^{i+1} \cdot T(n/3^{i+1}) + 29n \frac{1-(2/3)^{i+1}}{1-2/3} \end{aligned}$$

Για $i = \log_3 n - 1$:

$$\begin{aligned} T(n) &= 2^{\log_3 n} \cdot T(1) + 29n \cdot \frac{1-(2/3)^{\log_3 n}}{1/3} = 7 \cdot n^{\log_3 2} + 87n \cdot (1 - n^{\log_3 2/3}) \\ &= 7 \cdot n^{\log_3 2} + 87n \cdot (1 - n^{\log_3 2 - 1}) = 7 \cdot n^{\log_3 2} + 87n - 87n^{\log_3 2} \\ &= 87n - 80n^{\log_3 2} = \Theta(n) \end{aligned}$$

(β)

1. Κατ' αρχάς να πούμε ότι δεν μπορούμε να χρησιμοποιήσουμε το Master Theorem γιατί η $n/\log n$ είναι ασυμπτωτικά μικρότερη της n **αλλά όχι πολυωνυμικά!** Οπότε:

   $$T_1(n) = 5 \cdot T_1(n/5) + (x+1) \cdot n/\log n = 5 \cdot T_1(n/5) + (x+1)\log_5 2 \cdot n/\log_5 n$$

   Αν θεωρήσουμε για απλότητα ότι το n είναι δύναμη του 5, δηλ. ότι $n = 5^m$ για κάποιον ακέραιο m, τότε:

   $$T_1(5^m) = 5 \cdot T_1(5^{m-1}) + (x+1)\log_5 2 \cdot \frac{5^m}{m}$$

   Αν θέσουμε $T_1(m) = S(m)$, έχουμε ότι:

   $$\begin{aligned} S(m) &= 5S(m-1) + (x+1)\log_5 2 \cdot \frac{5^m}{m} \\ &= 5^2 S(m-2) + (x+1)\log_5 2 \cdot \frac{5^m}{m-1} + (x+1)\log_5 2 \cdot \frac{5^m}{m} = \dots \\ &= 5^i S(m-i) + (x+1)\log_5 2 \cdot 5^m \sum_{k=0}^{i-1} \frac{1}{m-k} \\ &\overset{i=m}{=} 5^m S(0) + (x+1)\log_5 2 \cdot 5^m \sum_{k=1}^{m} \frac{1}{i} \\ &= 5^m S(0) + (x+1)\log_5 2 \cdot 5^m (\ln m + \Theta(1)) \\ &= 5^m \Theta(1) + (x+1)\log_5 2 \cdot 5^m (\ln m + \Theta(1)) \end{aligned}$$

   Με αντίστροφη αντικατάσταση, έχουμε ότι:

   $$T_1(n) = n\Theta(1) + (x+1)\log_5 2 \cdot n(\ln\log n + \Theta(1)) \Rightarrow T_1(n) = \Theta(n \log\log n)$$

2. Εφαρμόζεται η τρίτη περίπτωση του Master Theorem, και έχουμε ότι $T_2(n) = \Theta(n^{5/2})$.

3. Έχουμε ότι:

$$
\begin{aligned}
T_3(n) &= T_3(n-1) + \frac{x+1}{n} = T_3(n-2) + \frac{x+1}{n-1} + \frac{x+1}{n} \\
&= T_3(n-3) + \frac{x+1}{n-2} + \frac{x+1}{n-1} + \frac{x+1}{n} = \dots \\
&= T_3(n-i) + (x+1)\sum_{k=0}^{i-1} \frac{1}{n-k} \overset{i=n-1}{=} T_3(1) + (x+1)\sum_{k=0}^{n-2} \frac{1}{n-k} \\
&= T_3(1) + (x+1)\sum_{k=2}^{n} \frac{1}{k} = \Theta(1) + \Theta(\log n) \\
&= \Theta(\log n)
\end{aligned}
$$

Μόνο οι δύο πρώτες χρησιμοποιούν "διαίρει-και-βασίλευε" αφού σπάνε το αρχικό πρόβλημα σε επιμέρους, μικρότερου(υποπολλαπλάσιου) μεγέθους.

Άσκηση 3: Βρείτε το Κάλπικο Νόμισμα

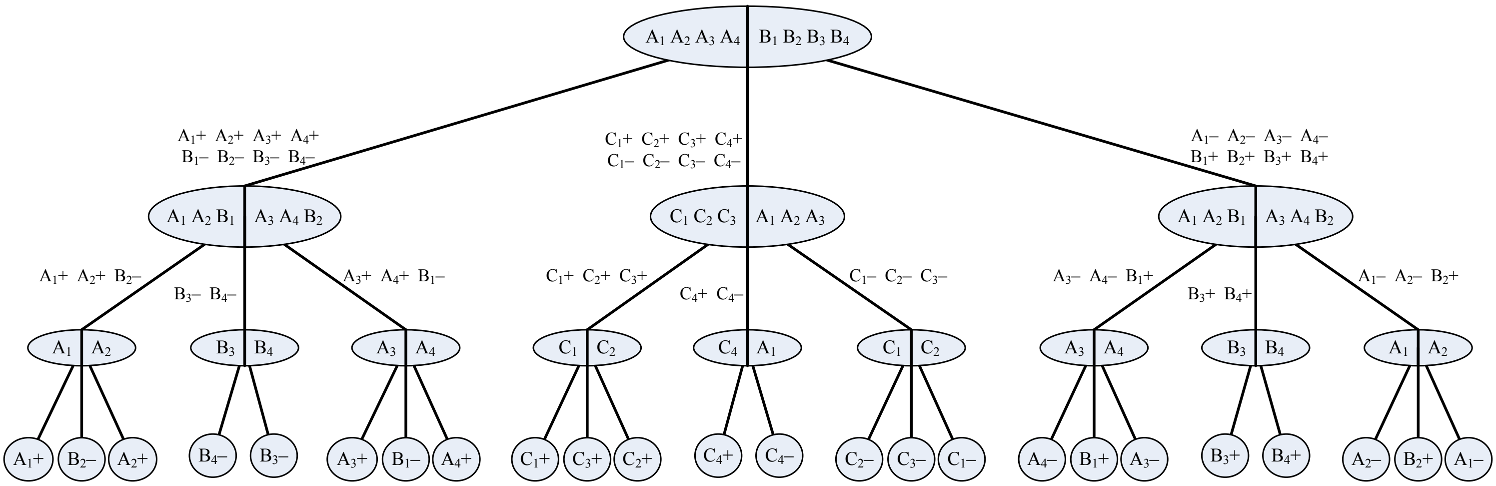
Κάθε "αλγόριθμος" που χρησιμοποιεί w ζυγίσεις για να εντοπίσει ένα κάλπικο ανάμεσα σε n νομίσματα μπορεί να αναπαρασταθεί με ένα τριαδικό δέντρο ζυγίσεων ύψους w το οποίο έχει τουλάχιστον $2n$ φύλλα, δύο για κάθε νόμισμα i, που αντιστοιχούν στα ενδεχόμενα το νόμισμα i να είναι κάλπικο και ελαφρύτερο από τα κανονικά (συμβολίζεται με $i-$) και το νόμισμα i να είναι κάλπικο και βαρύτερο από τα κανονικά (συμβολίζεται με $i+$). Το δέντρο είναι τριαδικό γιατί κάθε ζύγιση μπορεί να έχει τρία αποτελέσματα: η ζυγαριά γέρνει αριστερά, ισορροπεί, ή γέρνει δεξιά.

Αφού ένα τριαδικό δέντρο ύψους w έχει το πολύ 3^w φύλλα, ο μέγιστος αριθμός n_w νομισμάτων ανάμεσα στα οποία μπορούμε να εντοπίσουμε ένα κάλπικο νόμισμα (και αν αυτό είναι ελαφρύτερο ή βαρύτερο από τα κανονικά) με w ζυγίσεις είναι $n_w \leq (3^w - 1)/2$. Συνεπώς για 12 νομίσματα χρειαζόμαστε τουλάχιστον 3 ζυγίσεις. Ο πιο γνωστός "αλγόριθμος" για 12 νομίσματα με 3 ζυγίσεις φαίνεται στο Σχήμα 1. Στη βιβλιογραφία αναφέρονται και αρκετοί άλλοι, ένας μάλιστα καθορίζει τις ζυγίσεις που θα κάνει εξ΄αρχής, χωρίς αυτές να εξαρτώνται από τα αποτελέσματα των προηγούμενων ζυγίσεων, δείτε στην διεύθυνση:

`http://www.sciencedirect.com/science?_ob=ArticleURL&_udi=B6V0F-483SW0V-4&_user=83473&_rdoc=1&_fmt=&_orig=search&_sort=d&_docanchor=&view=c&_acct=C000059671&_version=1&_urlVersion=0&_userid=83473&md5=48118cfe50b4a4bb78cc17d780e7e8cf`

Ο μέγιστος αριθμός νομισμάτων n_2 ανάμεσα στα οποία μπορούμε να εντοπίσουμε το κάλπικο νόμισμα (και αν αυτό είναι ελαφρύτερο ή βαρύτερο από τα κανονικά) με 2 ζυγίσεις είναι $n_2 = 3$. Ο λόγος που $n_2 < 4$ είναι ότι αν στην πρώτη ζύγιση συμμετέχουν όλα τα 4 νομίσματα, έχουμε μόνο δύο ενδεχόμενα αποτελέσματα (η ζυγαριά δεν μπορεί να ισορροπήσει). Αν στην πρώτη ζύγιση συμμετέχουν μόνο δύο νομίσματα και η ζυγαριά ισορροπήσει, δεν μπορούμε με μία μόνο ζύγιση να βρούμε ποιό από τα άλλα δύο νομίσματα είναι κάλπικο και αν αυτό είναι ελαφρύτερο ή βαρύτερο από τα κανονικά. Μπορούμε βέβαια να εντοπίσουμε το κάλπικο ανάμεσα σε 4 νομίσματα με 2 μόνο ζυγίσεις είτε αν γνωρίζουμε ένα γνήσιο νόμισμα από τα 4 είτε αν έχουμε 3 επιπλέον νομίσματα που γνωρίζουμε ότι είναι γνήσια (δείτε π.χ. τον μεσαίο κλάδο στο Σχήμα 1).

Παρόμοια αποδεικνύουμε ότι μέγιστος αριθμός νομισμάτων n_3 (αντ. n_4) ανάμεσα στα οποία μπορούμε να εντοπίσουμε το κάλπικο νόμισμα (και αν αυτό είναι ελαφρύτερο ή βαρύτερο από τα



Σχήμα 1. Ένας "αλγόριθμος" που με 3 ζυγίσεις βρίσκει το κάλπικο ανάμεσα σε 12 νομίσματα και αν αυτό είναι ελαφρύτερο ή βαρύτερο. Ο "αλγόριθμος" χωρίζει αρχικά τα νομίσματα σε 3 τετράδες: την τετράδα A με τα νομίσματα A_1, A_2, A_3, A_4, την τετράδα B με τα νομίσματα B_1, B_2, B_3, B_4, και την τετράδα C με τα νομίσματα C_1, C_2, C_3, C_4. Η αναπαράσταση του αλγόριθμου βασίζεται σε ένα τριαδικό δέντρο συγκρίσεων. Κάθε κόμβος (εκτός από τα φύλλα) δηλώνει ότι τα νομίσματα στα αριστερά ζυγίζονται με τα νομίσματα στα δεξιά (π.χ. η ρίζα δηλώνει ότι τα νομίσματα της τετράδας A ζυγίζονται με αυτά της τετράδας B). Αν η ζυγαριά γείρει αριστερά (αντ. δεξιά), ακολουθείται ο αριστερός (αντ. δεξιός) κλάδος, και αν η ζυγαριά ισορροπήσει, ακολουθείται ο μεσαίος κλάδος. Σε κάθε κλάδο σημειώνονται τα πιθανά ενδεχόμενα με βάση την διαθέσιμη πληροφορία από τις ζυγίσεις. Η αναφορά σε ένα νόμισμα δηλώνει ότι αυτό μπορεί να είναι το κάλπικο, και το $+$ (αντ. $-$) δηλώνει ότι θα είναι βαρύτερο (αντ. ελαφρύτερο) από τα κανονικά (π.χ. τα ενδεχόμενα που αντιστοιχούν στον ψηλότερο αριστερό κλάδο είναι κάποιο από τα νομίσματα της τετράδας A να είναι κάλπικο και βαρύτερο από τα κανονικά ή κάποιο από τα νομίσματα της τετράδας B να είναι κάλπικο και ελαφρύτερο από τα κανονικά). Περιορίζοντας διαδοχικά τα ενδεχόμενα σε κάθε κλάδο, καταλήγουμε σε μοναδικό αποτέλεσμα σε κάθε φύλλο.

κανονικά) με 3 (αντ. 4) ζυγίσεις είναι $n_3 = 12$ (αντ. $n_4 = 39$). Για να εντοπίσουμε το κάλπικο νόμισμα ανάμεσα σε 39 νομίσματα με 4 ζυγίσεις, χωρίζουμε αρχικά τα νομίσματα σε 3 ομάδες των 13 νομισμάτων και εργαζόμαστε με την λογική του Σχήματος 1.

Γενικότερα, αποδεικνύεται ότι ο μέγιστος αριθμός νομισμάτων n_w ανάμεσα στα οποία μπορούμε να εντοπίσουμε το κάλπικο νόμισμα (και αν αυτό είναι ελαφρύτερο ή βαρύτερο από τα κανονικά) με w ζυγίσεις είναι $n_w = (3^w - 3)/2$.

Άσκηση 4: Μέτρηση Αντιστροφών

Η ιδέα είναι ίδια με αυτή της mergesort. Έστω ότι έχουμε μία ακολουθία με n στοιχεία. Επαγωγικά υποθέτουμε ότι το πρόβλημα έχει λυθεί για την υποακολουθία A με τα στοιχεία των $n/2$ πρώτων θέσεων του πίνακα και για την υποακολουθία B με τα στοιχεία των $n/2$ τελευταίων θέσεων του πίνακα. Επίσης υποθέτουμε ότι τα στοιχεία των υποακολουθιών A και B είναι ταξινομημένα. Σχηματικά έχουμε:

$a_1, a_2, \ldots, a_{\frac{n}{2}}$ $a_{(\frac{n}{2}+1)}, \ldots, a_n$

```
--------------            ---------------
r_A αντίστροφα ζεύγη      r_B αντίστροφα ζεύγη
και λίστα ταξινομημένη    και λίστα ταξινομημένη
```

Η λύση θα είναι $r = r_A + r_B + r_{AB}$, όπου r_{AB} το πλήθος των αντίστροφων ζευγών (a_i, a_j) με $i \le n/2$, $j > n/2$, και $a_i > a_j$.

Για να πετύχουμε πολυπλοκότητα $O(n \log n)$ πρέπει το merging να γίνει σε χρόνο $O(n)$ (θυμηθείτε το master theorem και την λύση της αναδρομικής εξίσωσης $T(n) = 2T(n/2) + \Theta(n)$.)

Κάνουμε το ίδιο merging με τη mergesort ώστε να ταξινομήσουμε τη λίστα, προσθέτοντας όμως ένα μετρητή r που κρατάει πόσα στοιχεία από την υποακολουθία A δεν έχουν ακόμη συγχωνευτεί. Κάθε φορά που ένα στοιχείο από την υποακολουθία B "συγχωνεύεται", το r_{AB} αυξάνεται κατά r, αφού αυτό "προηγείται" στην ταξινομημένη λίστα r στοιχείων που καταλαμβάνουν θέσεις στο πρώτο μισό του πίνακα. Μετά το merging επιστρέφεται η λύση $r_A + r_B + r_{AB}$, και η ακολουθία είναι ταξινομημένη. Ακολουθεί η διατύπωση αυτής της ιδέας σε ψευδοκώδικα:

```
Merge-and-Count(A, B)
  Διατήρησε δείκτη i στην A και δείκτη j στην B που αρχικά δείχνουν
      στα πρώτα στοιχεία τους.
  Διατήρησε μεταβλητή r_AB για τον αριθμό των αντιστροφών,
     με αρχική τιμή 0.
  While οι δύο λίστες δεν είναι κενές:
       Πρόσθεσε το μικρότερο από τα A[i] και B[j] στη λίστα εξόδου.
       If B[j] είναι το μικρότερο then
          Αύξησε το j κατά 1.
          Αύξησε το r_AB κατά το πλήθος των στοιχείων που απομένουν
              στη λίστα A
       If A[i] είναι το μικρότερο then
          Αύξησε το i κατά 1.
  EndWhile
  Όταν μια λίστα αδειάσει, πρόσθεσε τα υπόλοιπα στοιχεία της άλλης
       λίστας στην λίστα εξόδου.
  Επέστρεψε την ταξινομημένη λίστα και την μεταβλητή r_{AB}
```

Χρησιμοποιώντας την παραπάνω ρουτίνα ακολουθεί ο ψευδοκώδικας που λύνει το πρόβλημα:

```
Sort-and-Count(L)
 If η λίστα έχει ένα στοιχείο then δεν υπάρχουν αντιστροφές.
 Else
   Διαίρεσε τη λίστα σε δύο τμήματα:
 το A περιέχει τα πρώτα n/2 στοιχεία
 το B περιέχει τα υπόλοιπα στοιχεία
   (r_A, A)  = Sort-and-Count(A)
   (r_B, B)  = Sort-and-Count(B)
   (r_AB, L) = Merge-and-Count(A, B)
 Endif
 Return r_A+r_B+r_AB και την ταξινομημένη λίστα L
```

Άσκηση 5: Separators

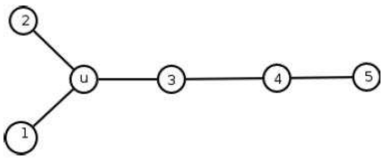
1. Θα δείξουμε στο ερώτημα αυτό ότι κάθε δέντρο έχει $\frac{1}{2}$-separator. Για την απόδειξη αυτή θα χρησιμοποιήσουμε επαγωγή. Για δέντρο μεγέθους 2, έχουμε τετριμμένη περίπτωση και βλέπουμε ότι και οι δύο κορυφές είναι $\frac{1}{2}$-separator για το δέντρο.

Έστω τώρα ότι υποθέτουμε ότι κάθε δέντρο T' μεγέθους n έχει $\frac{1}{2}$-separator. Θα δείξουμε ότι και κάθε δέντρο μεγέθους $n+1$ έχει $\frac{1}{2}$-separator. Έστω δέντρο μεγέθους $n+1$ (το ονομάζουμε T) από το οποίο αφαιρούμε ένα φύλλο ℓ, οπότε προκύπτει ένα δέντρο μεγέθους n, ας το ονομάσουμε αυτό T'. Έστω επίσης ότι ο $\frac{1}{2}$-separator του δέντρου T' είναι ο κόμβος u. Διακρίνουμε τώρα τις εξής περιπτώσεις:

- Μετά την προσθήκη του φύλλου ℓ, ο u παραμένει $\frac{1}{2}$-separator για το T.
- Μετά την προσθήκη του φύλλου ℓ, ο u δεν είναι $\frac{1}{2}$-separator για το T. Αυτό συμβαίνει γιατί το ℓ προστίθεται στη μεγαλύτερη συνεκτική συνιστώσα του $T' - \{u\}$ η οποία έχει $n/2$ κορυφές (και άρα το συνολικό πλήθος των κορυφών στις άλλες συνεκτικές συνιστώσες του $T' - \{u\}$ είναι το πολύ $(n-1)/2$). Τότε ο γείτονας του u στην συνεκτική συνιστώσα όπου προστέθηκε το φύλλο ℓ αποτελεί $\frac{1}{2}$-separator του T.

2. Έστω ότι κάποιο δέντρο έχει $\frac{1}{k}$-separator. Θα δείξουμε ότι στο δέντρο υπάρχει κόμβος με βαθμό τουλάχιστον k. Πράγματι, έστω ότι μετά την αφαίρεση του $\frac{1}{k}$-separator υπάρχουν m συνεκτικές συνιστώσες, η καθεμία με το πολύ n/k κορυφές. Αφού ο συνολικός αριθμός των κορυφών στις συνεκτικές συνιστώσες είναι $n-1$, πρέπει $m \geq k$. Για να προκύψουν όμως τόσες συνεκτικές συνιστώσες, πρέπει ο κόμβος που αφαιρείται να έχει αντίστοιχα μεγάλο βαθμό.

   Το αντίστροφο δεν ισχύει. Ένα αντιπαράδειγμα δέντρου που έχει κορυφή βαθμού 3, αλλά όχι και $\frac{1}{3}$ separator φαίνεται ακολούθως:

3. Το πρόβλημα λύνεται σε γραμμικό χρόνο ως εξής: Αρχικά σε γραμμικό χρόνο υπολογίζουμε το βαθμό όλων των κορυφών και τις κορυφές με βαθμό τουλάχιστον $x+3$, που είναι υποψήφιες για $1/(x+3)$-separators. Επιλέγουμε αυθαίρετα μία τέτοια κορυφή u και την θεωρούμε ως ρίζα του δέντρου και υποψήφιο $1/(x+3)$-separator. Υπολογίζουμε σε γραμμικό χρόνο το πλήθος των κορυφών σε κάθε υποδέντρο (αυτό γίνεται π.χ. με παραλλαγή της postorder), και αποθηκεύουμε την αντίστοιχη πληροφορία στη ρίζα κάθε υποδέντρου. Το u είναι $1/(x+3)$-separator αν όλες οι συνεκτικές συνιστώσες που προκύπτουν από την αφαίρεσή του έχουν το πολύ $n/(x+3)$ κορυφές. Αν υπάρχουν δύο ή περισσότερες συνιστώσες με περισσότερες από $n/(x+3)$ κορυφές, το δέντρο δεν έχει $1/(x+3)$-separator (αυτές βρίσκονται σε διαφορετικά υποδέντρα, και συνεπώς δεν μπορούν να "διασπαστούν" ταυτόχρονα με την αφαίρεση μόνο μίας κορυφής). Αν μία μόνο συνιστώσα έχει περισσότερες από $n/(x+3)$ κορυφές, τότε θεωρούμε ως νέα ρίζα του δέντρου και υποψήφιο $1/(x+3)$-separator την κορυφή u' αυτής της συνιστώσας που έχει βαθμό τουλάχιστον $x+3$ και βρίσκεται πλησιέστερα στο u. Παρατηρούμε ότι η πληροφορία για πλήθος των κορυφών σε κάθε υποδέντρο υπολογίζεται από την σχετική πληροφορία που αναφέρεται στη ρίζα u σε χρόνο ανάλογο του πλήθους των ακμών στο μονοπάτι από την u στην u' στο δέντρο. Η διαδικασία συνεχίζεται ενόσω η αφαίρεση της τρέχουσας ρίζας οδηγεί σε μοναδική συνιστώσα με περισσότερες από $n/(x+3)$ κορυφές που δεν περιέχει τις προηγούμενες ρίζες. Αν έτσι οδηγηθούμε σε μοναδική συνιστώσα με περισσότερες από $n/(x+3)$ κορυφές που περιέχει τις προηγούμενες ρίζες, το δέντρο δεν έχει $1/(x+3)$-separator.

   Ο χρόνος εκτέλεσης είναι συνολικά γραμμικός γιατί σε κάθε επανάληψη, η νέα ρίζα βρίσκεται σε κάποιο υποδέντρο της τρέχουσας ρίζας στο αρχικό δέντρο με ρίζα την u. Έτσι για κάθε ακμή του δέντρου, η πληροφορία για τον αριθμό των κορυφών στα αντίστοιχα υποδέντρα θα ενημερωθεί μία φορά το πολύ.

Άσκηση 6: Η Διάβαση του Ποταμού

(α) Η λύση του παραλείπεται.
(β) Παρατηρούμε ότι όταν ο βαρκάρης μετακινεί κάποια αντικείμενα στην άλλη μεριά μιας όχθης, τα αντικείμενα που μένουν στη μια όχθη θα πρέπει να είναι ασύμβατα μεταξύ τους, ώστε να μην υπάρχει κίνδυνος. Συνεπώς, κάθε φορά θα πρέπει να απομακρύνεται ένα σύνολο κόμβων, ώστε οι κόμβοι που απομένουν να μην έχουν καμία ακμή μεταξύ τους. Ενδεικτικά για κάποιες περιπτώσεις έχουμε (στη χωρητικότητα της βάρκας δεν προσμετρούμε τον βαρκάρη):

- *Αλυσίδα μεγέθους n:* Έστω ότι έχουμε την αλυσίδα $\{x_1, x_2, ..., x_n\}$. Αν το n είναι άρτιος αριθμός, ο βαρκάρης μπορεί να μεταφέρει όλα τα αντικείμενα με βάρκα χωρητικότητας $n/2$ με 3 διαδρομές: μεταφέρει πρώτα τα μισά (π.χ. τους κόμβους με περιττό δείκτη), γυρίζει μόνος του, και στη συνέχεια μεταφέρει τα υπόλοιπα. Όταν το n είναι περιττό, θα χρειαστεί βάρκα χωρητικότητας $(n-1)/2$ και 7 διαδρομές συνολικά: μεταφέρει πρώτα τα $x_2, x_4, \ldots x_{n-1}$, επιστρέφει μόνος του, μεταφέρει το αντικείμενο x_1, παίρνει όλα τα "άρτια" αντικείμενα μαζί του στην αριστερή όχθη, μετά μεταφέρει τα $x_3, x_5, \ldots, x_n$ στη δεξιά όχθη, επιστρέφει, και τελικά μεταφέρει όλα τα "άρτια" αντικείμενα και πάλι στη δεξιά όχθη.
- *Δακτύλιος μεγέθους n:* Το δρομολόγιο είναι παρόμοιο με αυτό της αλυσίδας άρτιου μεγέθους. Αν το n είναι περιττό, η βάρκα πρέπει να έχει μέγεθος $(n+1)/2$.
- *Αστέρι μεγέθους n:* Χρειάζονται $2n-3$ διαδρομές με βάρκα χωρητικότητας 2. Το "κεντρικό" αντικείμενο είναι μόνιμα φορτωμένο στη βάρκα, και κάθε φορά μεταφέρεται ένα "περιφερειακό" αντικείμενο.
- *Πλήρης γράφος μεγέθους n:* Χρειάζονται 3 διαδρομές με βάρκα χωρητικότητας $n-1$.
- *Πλέγμα διαστάσεων $\sqrt{n} \times \sqrt{n}$:* Για το πλέγμα αρκεί να αφαιρέσουμε τους κόμβους που βρίσκονται διαγώνια μεταξύ τους. Ενδεικτικά, από την πρώτη σειρά θα πάρουμε τον πρώτο, τρίτο, κ.ο.κ. κόμβους, από τη δεύτερη σειρά το δεύτερο, τέταρτο, κ.ο.κ. κόμβο, από την τρίτη σειρά τον πρώτο, τον τρίτο, κ.ο.κ. και αυτό μέχρι την τελευταία σειρά. Συνεπώς, η χωρητικότητα της βάρκας πρέπει να είναι $\lfloor n/2 \rfloor$.

(γ) Στην πρώτη διαδρομή μεταφέρουμε κάποια αντικείμενα στην απέναντι όχθη. Για να γίνει αυτό σύμφωνα με τους περιορισμούς του προβλήματος, πρέπει το επαγόμενο υπογράφημα ασυμβατοτήτων για τα αντικείμενα που έμειναν στην όχθη να αποτελεί ένα σύνολο ανεξαρτησίας. Άρα τα αντικείμενα που μεταφέρθηκαν πρέπει να αποτελούν ένα κάλυμμα κορυφών του γραφήματος (ας θυμηθούμε ότι ένα υποσύνολο κορυφών αποτελεί κάλυμμα κορυφών ανν οι υπόλοιπες κορυφές αποτελούν σύνολο ανεξαρτησίας). Έτσι για να υπάρχει λύση στο πρόβλημα, πρέπει η χωρητικότητα της βάρκας να είναι μεγαλύτερη ή ίση με το ελάχιστο κάλυμμα κορυφών του γραφήματος ασυμβατοτήτων.