ΕΠΛ 232 – Φροντιστήριο 2

Πρόβλημα 1

Έχετε £10 και θέλετε να τις επενδύσετε για n μήνες. Την πρώτη μέρα κάθε μήνα έχετε **μόνο μια** από τις παρακάτω τρεις επιλογές:

1. Να αγοράσετε ένα πιστοποιητικό αποταμίευσης από τη τράπεζα. Κατά το μήνα t θα σας στοιχίσει ένα δικαίωμα C_α(t) προς τη τράπεζα και μετά από ένα μήνα θα σας επιστρέψει το ποσό που καταθέσατε επί ένα συντελεστή A(t). Δηλαδή, αν το μήνα t καταθέσετε με αυτό τον τρόπο £K, μετά από ένα μήνα θα έχετε £(K-C_α(t))·A(t).
2. Να αγοράσετε ένα κρατικό ομόλογο αποταμίευσης από το χρηματιστήριο. Κατά το μήνα t θα σας στοιχίσει ένα δικαίωμα C_o(t) προς το χρηματιστήριο και μετά από δύο μήνες θα σας επιστρέψει το ποσό που καταθέσατε επί ένα συντελεστή O(t). Δηλαδή, αν το μήνα t κατεθέσετε με αυτό τον τρόπο £K, μετά από δύο μήνες θα έχετε £(K-C_o(t))·O(t).
3. Να μην αγοράσετε τίποτα.

Υποθέτοντας ότι γνωρίζετε τις συναρτήσεις C_α(t), C_o(t), A(t), O(t), να δώσετε αλγόριθμο ο οποίος να υπολογίζει το μέγιστο χρηματικό ποσό που μπορείτε να έχετε μετά από n μήνες και την αντίστοιχη ακολουθία από βέλτιστες επιλογές.

Χρησιμοποιούμε δυναμικό προγραμματισμό.
Έστω V(t), $0 \le t \le n$, το μέγιστο χρηματικό ποσό που μπορώ να έχω μετά από t μήνες. Προφανώς V(0)=10 και V(n) είναι το ζητούμενο.

Η ακολουθία από βέλτιστες επιλογές για τους πρώτους k μήνες είναι ένα διάνυσμα της μορφής:

$$C_k = \langle c_1, \ldots, c_k \rangle$$

του οποίου οι τιμές προέρχονται από το σύνολο $\{A, O, T\}$. Ζητούμενο είναι το C_n.

Χαρακτηρισμός της βέλτιστης λύσης

Λήμμα: Έστω το διάνυσμα βέλτιστων επιλογών

$$C_k = \langle c_1, \ldots, c_k \rangle$$

(ι) Αν c_k =Α ή Τ, τότε το

$$\langle c_1, \ldots, c_{k-1} \rangle$$

περιέχει τις βέλτιστες επιλογές για τους k-1 πρώτους μήνες.

(ιι) Αν c_k = Ο, τότε το

$$\langle c_1, \ldots, c_{k-2} \rangle$$

περιέχει τις βέλτιστες επιλογές για τους k-2 πρώτους μήνες.

Απόδειξη του (ι)

Υποθέτουμε πως όχι, δηλαδή ότι το

$$\langle c_1', \ldots, c_{k-1}' \rangle \neq \langle c_1, \ldots, c_{k-1} \rangle$$

περιέχει τις βέλτιστες επιλογές για τους k-1 πρώτους μήνες. Τότε το

$$\langle c_1', \ldots, c_{k-1}', c_k \rangle$$

περιέχει καλύτερες επιλογές για τους k πρώτους μήνες από εκείνες του C . Αντίφαση.

Η απόδειξη του (ιι) είναι όμοια.

□

Άρα

$$V(t) = \begin{cases} max((V(t-1) - C_A(t-1)) \cdot A(t-1), \\ \qquad V(t-1)), & \text{if } t = 1 \\ \\ max((V(t-1) - C_A(t-1)) \cdot A(t-1), \\ \qquad (V(t-2) - C_O(t-2)) \cdot O(t-2), \\ \qquad V(t-1)), & \text{if } t \neq 1 \end{cases}$$

Υπολογίζουμε το V(n) αρχίζοντας από τα V(1), V(2),…

Χρόνος Εκτέλεσης:

Πρόβλημα 2

Έχουμε στη διάθεση μας απεριόριστη ποσότητα n νομισμάτων όπου το i-οστό νόμισμα έχει αξία d_i. Να σχεδιασθεί αλγόριθμος ο οποίος, για οποιοδήποτε ακέραιο X επιτρέφει τον ελάχιστο αριθμό νομισμάτων με συνολική αξία X.

- Γνωρίζουμε ότι *στη γενική περίπτωση* το πιο πάνω πρόβλημα δεν μπορεί να λυθεί με τον προφανή άπληστο αλγόριθμο (π.χ. όταν έχουμε 4 νομίσματα αξίας 1, 10, 25, 100)

- Αφού όμως το πρόβλημα ικανοποιεί την ιδιότητα της βέλτιστης υποδομής μπορούμε να το λύσουμε με δυναμικό προγραμματισμό.

- Χρησιμοποιούμε ένα πίνακα c[1..n, 0..X], με μια σειρά για κάθε νόμισμα και μια στήλη για κάθε βάρος μεταξύ 0 και X. Στη θέση c[i,j] του πίνακα θα αποθηκεύουμε τον ελάχιστο αριθμό συλλογής νομισμάτων από νομίσματα 1..i αξίας j.

- Έτσι λύση στο πρόβλημα μας είναι η τιμή c[n,X].

- Αν j=0, τότε c[i,0]= 0. Διαφορετικά, η ελάχιστη συλλογή νομισμάτων αξίας j που περιέχει μόνο τα νομίσματα 1...i μπορεί
  a. είτε να μην περιέχει το νόμισμα i, οποτε
     c[i,j] = c[i-1,j],
  b. είτε να το περιέχει, οπότε
     c[i,j] = c[i,j-d_i] + 1

  Έτσι έχουμε τη σχέση

$$c[i,j] = \begin{cases} c[i-1,j], & \text{if } j < d_i \\ min(c[i-1,j], c[i,j-d_i]) + 1, & \text{otherwise} \end{cases}$$

- Υπολογίζουμε το V[n,W], ξεκινώντας από το V[1,0],…,V[1,W] και προχωρώντας στο V[2,_], V[3,_],…

Παράδειγμα

Έστω Χ=8 και έχουμε στη διάθεση μας 3 αξίας 1, 4 και 6. Εκτέλεση του αλγόριθμου δίνει τον πιο κάτω πίνακα

i / j	0	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	7	8
2	0	1	2	3	1	2	3	4	2
3	0	1	2	3	1	2	1	2	2

Εργασία 1

Να γράψετε αναλυτικά τον αλγόριθμο λύσης του προβλήματος και να αναλύσετε το χρόνο εκτέλεσής του.

Λύση με αλγόριθμο οπισθοδρόμησης

- Μία επιλογή είναι i-υποσχόμενη αν
  1. περιέχει i νομίσματα και
  2. η αξία της δεν ξεπερνά τη τιμή X.

- Θεωρώντας ότι ξεκινούμε με το 0-υποσχόμενο διάνυσμα είναι δυνατό να κτίσουμε όλα τα 1-υποσχόμενα διανύσματα, 2-υποσχόμενα διανύσματα και ούτω καθεξής. Με δεδομένο ένα i-υποσχόμενο διάνυσμα υπάρχουν τρείς περιπτώσεις
  1. είτε αυτό μπορεί να γίνει (i+1)-υποσχόμενο διάνυσμα με την προσθήκη ενός νομίσματος,
  2. είτε η συλλογή έχει αξία X,
  3. είτε έχουμε φτάσει σε αδιέξοδο, δηλαδή σε μια συλλογή στη οποία οποιαδήποτε προσθήκη νομισμάτων συνεπάγεται συγκέντρωση αξίας > X.

- Στην περίπτωση 2
  1. σημειώνουμε τον αριθμό νομισμάτων της συλλογής (για μελλοντικές συγκρίσεις) και
  2. εφαρμόζουμε οπισθοδρόμηση.

- Στην περίπτωση 3
  εφαρμόζουμε οπισθοδρόμηση.

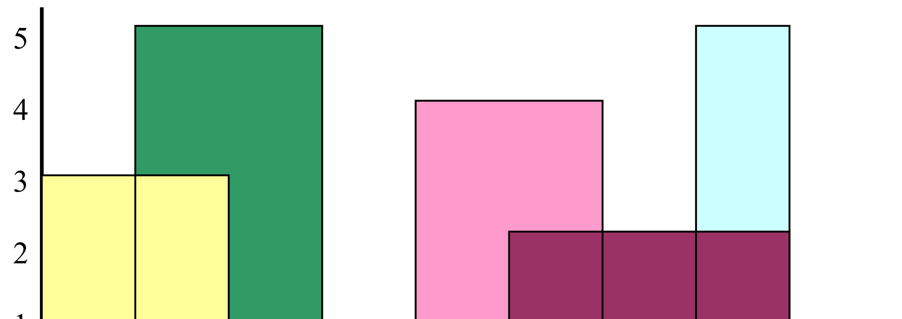
- Όταν εξερευνήσουμε όλες τις περιπτώσεις επιστρέφουμε την ελάχιστη τιμή από αυτές που έχουμε σημειώσει.

Εργασία 2

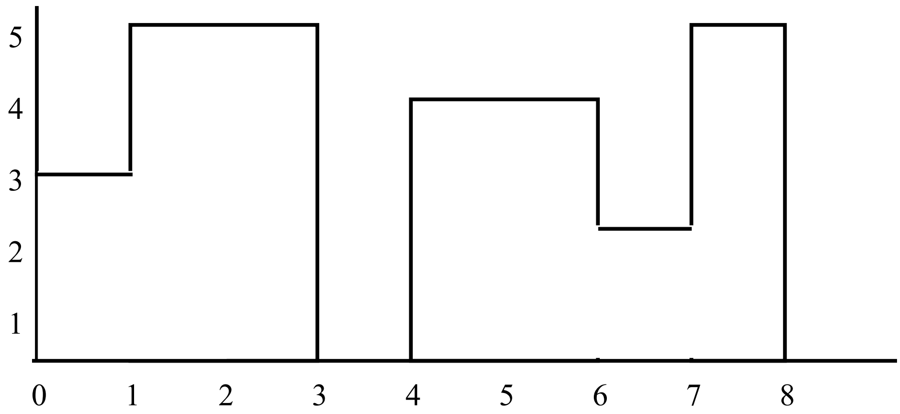
Να γράψετε τον πιο πάνω αλγόριθμο οπισθοδρόμησης και να τον συγκρίνετε με τον αλγόριθμο δυναμικού προγραμματισμού για το ίδιο πρόβλημα.

Πρόβλημα 3

Σας δίνονται οι ακριβείς τοποθεσίες και διαστάσεις (ορθογώνιων) κτιρίων κάποιας πόλης και θέλετε να σχεδιάσετε (σε δύο διαστάσεις) την κορυφογραμμή αυτών των κτιρίων, χωρίς να φαίνονται οποιεσδήποτε κρυμμένες γραμμές. Συγκεκριμένα υποθέστε ότι όλα τα κτίρια βρίσκονται στην ίδια γραμμή και ένα κτίριο Κ σας δίνεται ώς την τριάδα (Α,Δ,Υ) όπου Α και Δ είναι οι αριστερή και δεξιά συνιστώσα του Κ, αντίστοιχα, και Υ είναι το ύψος του. Η κορυφογραμμή ενός συνόλου από κτίρια είναι μια ακολουθία από x-συνιστώσες και τα ύψη που τις συνδέουν, προχωρώντας από τα αριστερά προς τα δεξιά. Για παράδειγμα τα πιο κάτω κτίρια



αντιστοιχούν στο δεδομένο εισόδου {(0,2,3), (1,3,5), (4,6,4), (5,8,2), (7,8,5)}, και η ζητούμενη κορυφογραμμή είναι η 0,3,1,5,3,0,4,4,6,2,7,5,8,0:



i. Να σχεδιάσετε αλγόριθμο, ο οποίος με δεδομένο εισόδου μια κορυφογραμμή n κτιρίων και μία κορυφογραμμή m κτιρίων, βρίσκει και επιστρέφει την κορυφογραμμή των n+m κτιρίων σε χρόνο O(n+m).

ii. Να σχεδιάσετε “διαίρει και βασίλευε” αλγόριθμο ο οποίος υπολογίζει την κορυφογραμμή n κτιρίων σε χρόνο O(n lg n).