Εισαγωγή στον Σχεδιασμό και Ανάλυση Αλγορίθμων

Τμήμα Μαθηματικών ΕΚΠΑ

Εξέταση Σεπτεμβρίου 2018

1. [7 μ.] Έστω διάνυσμα v με ακέραιες τιμές (μπορεί να περιέχει και *αρνητικές* τιμές). Γράψτε αλγόριθμο δυναμικού προγραμματισμού που να υπολογίζει το μέγιστο από τα αθροίσματα τιμών των *συνεχόμενων* υποδιανυσμάτων του v. Εξηγήστε πώς προκύπτει η αναδρομική σχέση στην οποία στηρίζεται ο αλγόριθμος και υπολογίστε το κόστος του σε χρόνο και μνήμη.

Λύση:

Το μέγιστο άθροισμα C_j που προκύπτει από τα όλα τα υποδιανύσματα του v μέχρι τη θέση j είναι

- είτε το $C_{j-1} + v_j$
- είτε το v_j.

Η αναδρομική σχέση:

$$C_j = \max\{C_{j-1} + v_j, v_j\}, \quad C_1 = v_1.$$

Ο αλγόριθμος:

Input: $v[1 \dots n]$
Output: s μέγιστο άθροισμα
  $C[1] = v[1]$
  for $j = 2$ **to** n **do**
    $C[j] = \max(C[j-1] + v[j], v[j])$
  return $s = \max(C)$

Ο αλγόριθμος είναι $\Theta(n)$ ως προς χρόνο και χώρο.

2. [3 μ.] Σε "σκακιέρα" $m \times n$ θέσεων τοποθετούνται κέρματα *ίσης αξίας*, με *το πολύ ένα* κέρμα σε κάθε θέση. Ένα ρομπότ ξεκινάει από την *πάνω αριστερή* θέση, και πρέπει να καταλήξει στην *κάτω δεξιά* θέση, συλλέγοντας όσα περισσότερα κέρματα μπορεί. Κάποιες από τις ενδιάμεσες θέσεις είναι *μη-προσπελάσιμες*. Σε κάθε βήμα το ρομπότ μετακινείτε είτε μια θέση *δεξιά* είτε μια θέση *κάτω* από την θέση στην οποία βρίσκεται. Το ρομπότ *παίρνει πάντα* το νόμισμα που θα βρει σε μια θέση. Γράψτε αλγόριθμο δυναμικού προγραμματισμού που να υπολογίζει τον μέγιστο αριθμό κερμάτων που μπορεί να συλλέξει το ρομπότ.

Λύση:

Αρκεί να θέσουμε την τιμή $-\infty$ στις μη-προσπελάσιμες θέσεις.

Input: Πίνακας $C[1 \ldots m, 1 \ldots n]$ με στοιχεία $0, 1$ και $-\infty$.
Output: $F[m, n]$ - max αριθμός κερμάτων στη θέση (m, n).

```
F[1,1] = C[1,1]
for j = 2 to n do
    F[1,j] = F[1,j-1] + C[1,j]
for i = 2 to m do
    F[i,1] = F[i-1,1] + C[i,1]
    for j = 2 to n do
        F[i,j] = max(F[i-1,j], F[i,j-1] + C[i,j])
return F[m,n]
```