από (6) : $\mu_3 = 0$.

(1) $\Rightarrow \frac{1}{2} + 1 - \mu_1 = 0 \Rightarrow \mu_1 = \frac{3}{2}$

(2) $\Rightarrow \mu_4 = 1 \xRightarrow{(7)} x_2 = 0$

κ' ελέγχουμε αν ικανοποιούνται όλες οι υπόλοιπες εξισώσεις.

Δευτέρα 17/12/2012

Δυναμικός προγραμματισμός

π.χ. 1

Έστω μια ακολουθία n πραγματικών αριθμών $\alpha_1, \ldots, \alpha_n$. Θέλουμε να βρούμε την υπακολουθία $\alpha_i, \alpha_{i+1}, \ldots, \alpha_j$ διαδοχικών αριθμών με το μέγιστο άθροισμα

$\left(i<j \; , \; i,j \in \{1, \ldots, n\} \; , \; \underbrace{\alpha_1, \ldots, \alpha_i, \ldots, \alpha_j, \ldots, \alpha_n} \right)$

$M(j) :=$ μέγιστο άθροισμα όλων των δυνατών υπακολουθιών που τελειώνουν στο j.

Οπότε, $M(j) = \max \{ M(j-1) + \alpha_j \, , \, \alpha_j \}$.

$\left(\ldots \ldots, \underbrace{\alpha_{j-1}, \alpha_j}, \ldots \right.$
συνεχίζουμε το βέλτιστο προηγούμενο παράθυρο (υπακολουθία)
ή ξεκινάμε ένα καινούριο.)

Θέλουμε να υπολογίσουμε το $M(n)$.

Υπολογιστικό κόστος $O(n)$
(έχουμε να λύσουμε n προβλήματα $(j=1, \ldots, n)$ που το καθένα θέλει
χρόνο 1.)

π.χ. 2.

Έστω n κέρματα με αξία $v(1) < v(2) < \ldots < v(n)$, με $v(1) = 1$.
κ' $v(i) \in \mathbb{Z}_+$, $i \in \{1, \ldots, n\}$

Θέλουμε να χωρίσουμε σε κέρματα ένα ποσό C, με όσο το δυνατό λιγότερα.

Έστω M_j ο ελάχιστος αριθμός κερμάτων που απαιτείται για να συμπληρώσουμε το ποσό j

$$M(j) = \min_i M(j - v(i)) + 1 \quad , i \in \{1, \ldots, n\}.$$

Υπολογιστικός χρόνος $O(nC)$
Λύνουμε C προβλήματα, που το καθένα απαιτεί χρόνο n.

π.χ. 3

Έστω μια ακολουθία n αριθμών $\alpha_1, \ldots, \alpha_n$.
Να βρεθεί η μέγιστη σε μήκος υπακολουθία τα στοιχεία της οποίας είναι αύξοντα (όχι υποχρεωτικά διαδοχικά).

Έστω $L(j)$ ~~μιας τέτοιας~~ (~~ελάχιστης~~) το μήκος ακολουθίας μέχρι τη θέση j.

$$L(j) = \max_{\substack{i<j \\ \alpha_i < \alpha_j}} L(i) + 1$$

κ' λύνουμε το πρόβλημα $\max_j L(j)$.

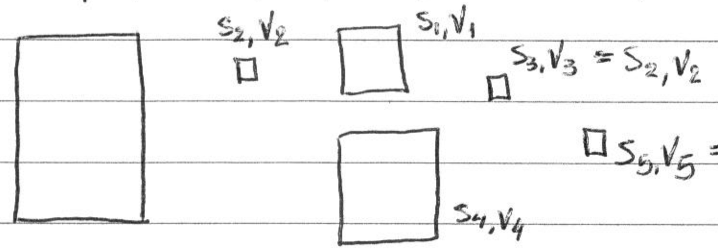
Υπολογιστικός χρόνος $O(n^2)$
Έχουμε n προβλήματα που το καθένα απαιτεί χρόνο n.

<u>π.χ. 4</u>

Έχουμε n αντικείμενα με μέγεθος $s_i \in \mathbb{Z}_+$ κ' αξία v_i κ' θέλουμε να τα τοποθετήσουμε σε σακίδιο χωρητικότητας $C \in \mathbb{Z}_+$, μεγιστοποιώντας την αξία.

<u>Περίπτωση 1</u>

Μπορούμε να έχουμε όμοια αντικείμενα (ως προς το μέγεθος κ' την αξία)



Έστω $M(j)$ η μέγιστη αξία για σακίδιο χωρητικότητας j.

$$M(j) = \max\{\underbrace{M(j-1)}_{\substack{\text{Αν η λύση για}\\ \text{χωρητικότητα } j\\ \text{δεν είναι διαφορετική}\\ \text{εκείνης για } j-1}}, \underbrace{\max_i \{M(j-s_i)+v_i\}}_{\substack{\text{Αν υπάρχει κάποιο αντικείμενο } i\\ \text{που γεμίζει πλήρως το σακίδιο}\\ \text{μέχρι τη χωρητικότητα } j.}}\}$$

Λύνουμε για $j=1,\dots,C$ κ' σε κάθε βήμα παίρνουμε $\max_i$, δηλαδή n πράξεις.
Οπότε συνολικά έχουμε <u>υπολογιστικό χρόνο</u> $O(nC)$