Εισαγωγή στους Αλγορίθμους

Ενότητα 11η

Διδάσκων
Χρήστος Ζαρολιάγκης
Καθηγητής
Τμήμα Μηχανικών Η/Υ & Πληροφορικής
Πανεπιστήμιο Πατρών
Email: zaro@ceid.upatras.gr

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.

- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.

- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.

- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σκοποί ενότητας

- Εισαγωγή στο Δυναμικό Προγραμματισμό
- Περιγραφή και ανάλυση του σταθμισμένου χρονοπρογραμματισμού διαστημάτων

Περιεχόμενα ενότητας

- Δυναμικός Προγραμματισμός
- Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων

Δυναμικός Προγραμματισμός

- Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων

Αλγοριθμικά Μοντέλα

Απληστία. Χτίσε μια λύση σταδιακά, βελτιστοποιώντας μυωπικά κάποιο τοπικό κριτήριο.

Διαίρει και βασίλευε. Διάσπασε ένα πρόβλημα σε δύο υποπροβλήματα, λύσε κάθε υποπρόβλημα ανεξάρτητα, και συνδύασε τις λύσεις των υποπροβλημάτων για να δημιουργήσεις την λύση του αρχικού προβλήματος.

Δυναμικός προγραμματισμός. Διάσπασε ένα πρόβλημα σε μια σειρά από επικαλυπτόμενα υποπροβλήματα, και δόμησε σωστές λύσεις για όλο και μεγαλύτερα υποπροβλήματα.

Ιστορία Δυναμικού Προγραμματισμού

Bellman. Ήταν πρωτοπόρος της συστηματικής μελέτης του δυναμικού προγραμματισμού την δεκαετία του 1950.

Ετυμολογία.

- Δυναμικός προγραμματισμός = σχεδιασμός με την πάροδο του χρόνου.
- Η τότε κυβέρνηση των ΗΠΑ ήταν αντίθετη με την έρευνα στο πεδίο των μαθηματικών.
- Ο Bellman αναζήτησε ένα εντυπωσιακό όνομα για να αποφύγει την αντιπαράθεση.
  - "είναι αδύνατον να χρησιμοποιηθεί η λέξη 'δυναμικός' με υποτιμητική έννοια"
  - "κάτι στο οποίο δεν θα μπορούσε να διαφωνήσει ούτε ένα μέλος του κογκρέσου"

Αναφορά: Bellman, R. E. *Eye of the Hurricane, An Autobiography.*

Εφαρμογές δυναμικού προγραμματισμού

Περιοχές.

- Βιοπληροφορική.
- Θεωρία του ελέγχου.
- Θεωρία της πληροφορίας.
- Ερευνητικές δραστηριότητες.
- Επιστήμη υπολογιστών: θεωρία, γραφικά, τεχνητή νοημοσύνη, συστήματα, ….

Μερικοί διάσημοι αλγόριθμοι δυναμικού προγραμματισμού.

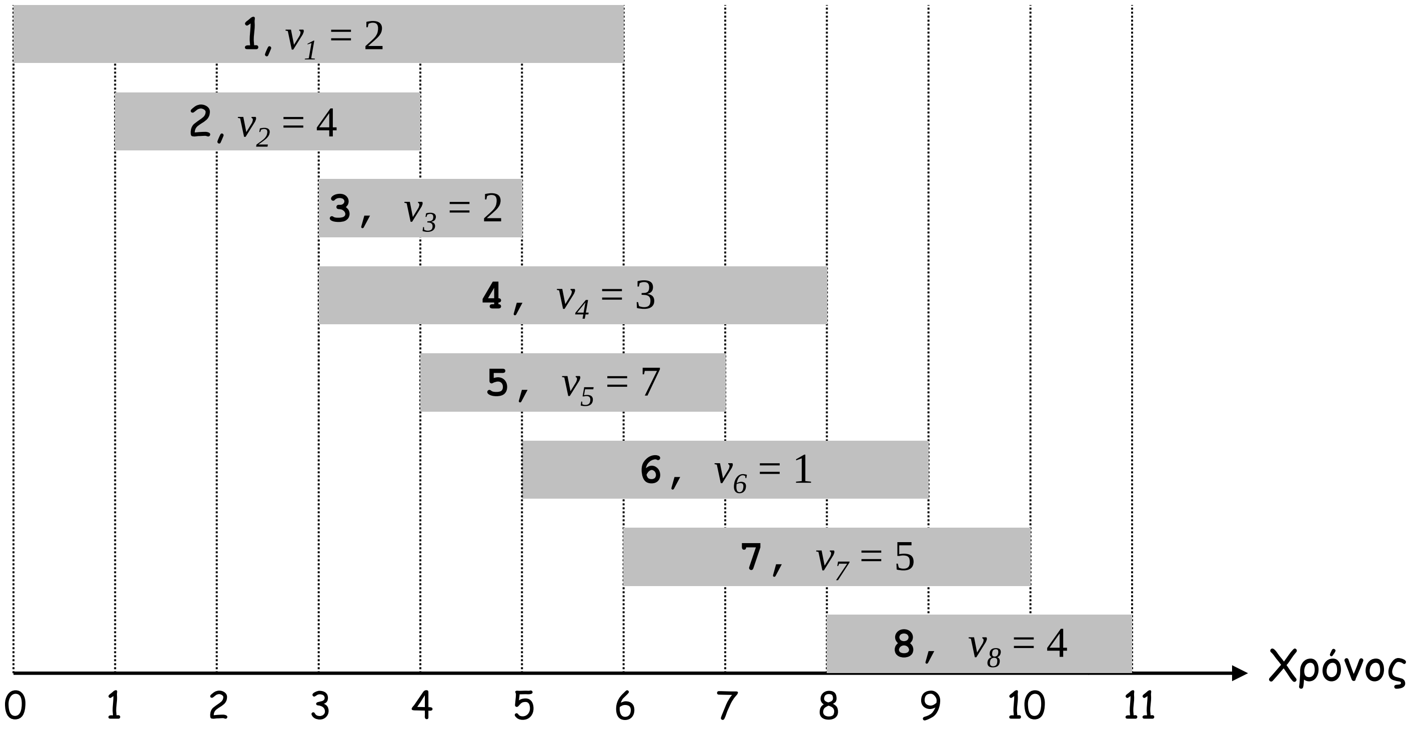
- Viterbi για κρυμμένα Μαρκοβιανά μοντέλα.
- Unix diff για σύγκριση δύο αρχείων.
- Smith-Waterman για ευθυγράμμιση ακολουθιών.
- Bellman-Ford για δρομολόγηση συντομότερης διαδρομής σε δίκτυα.
- Cocke-Kasami-Younger για ανάλυση γραμματικών χωρίς συμφραζόμενα.

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων

Πρόβλημα σταθμισμένου χρονοπρογραμματισμού διαστημάτων.

- Σύνολο αιτημάτων ή εργασιών $\{1, \ldots, n\}$
- Το αίτημα j ξεκινά την στιγμή s_j, τελειώνει την στιγμή f_j, και έχει βαρύτητα v_j
- Δύο αιτήματα είναι συμβατά αν δεν επικαλύπτονται.
- Στόχος: εύρεση υποσυνόλου $S \subseteq \{1, \ldots, n\}$ συμβατών αιτημάτων μέγιστης βαρύτητας $\Sigma_{i \in S}\, v_i$.

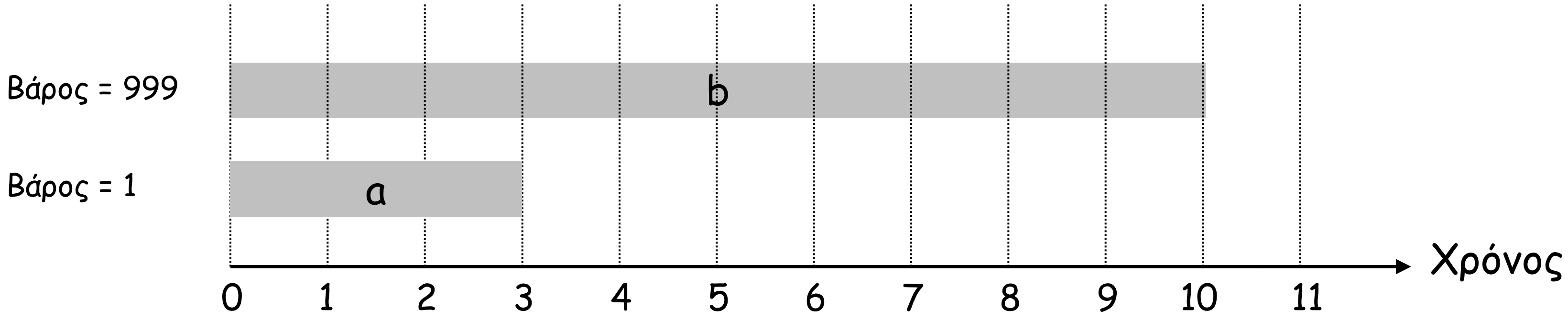


Ανασκόπηση Μη-σταθμισμένου Χρονοπρογραμματισμού Διαστημάτων

Ανασκόπηση. Ο άπληστος αλγόριθμος λειτουργεί αν όλες οι βαρύτητες είναι 1.

- Θεωρείστε τα αιτήματα σε αύξουσα σειρά ως προς το χρόνο λήξης.
- Προσθέστε το αίτημα στο υποσύνολο αν είναι συμβατό με τα αιτήματα που έχουν ήδη επιλεχθεί.

Παρατήρηση. Ο άπληστος αλγόριθμος μπορεί να αποτύχει θεαματικά αν επιτραπούν αυθαίρετες βαρύτητες.

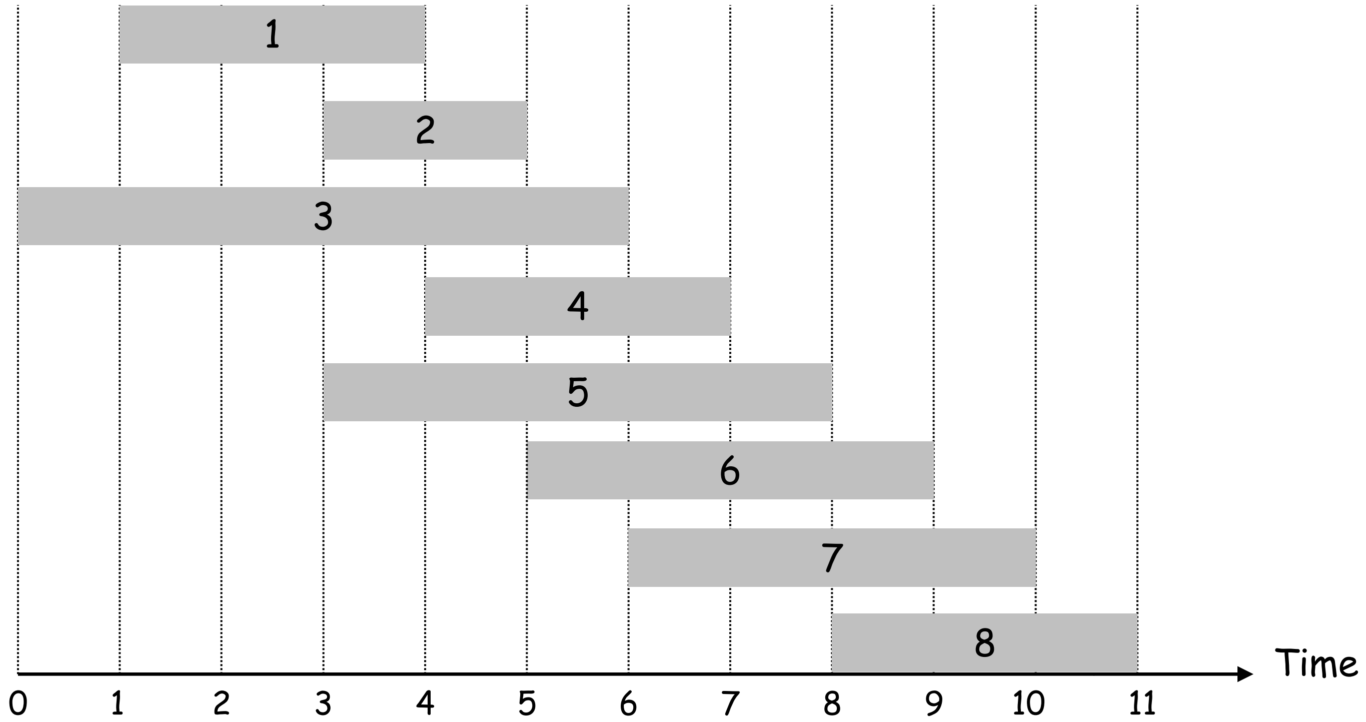


Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων

Διατάσσουμε τα αιτήματα σύμφωνα με το χρόνο λήξης τους: $f_1 \leq f_2 \leq \ldots \leq f_n$

p(j) = μεγαλύτερος δείκτης i < j τέτοιος ώστε το αίτημα i να είναι συμβατό με το j.

Παράδειγμα: p(8) = 5, p(7) = 3, p(2) = 0.



Δυναμικός Προγραμματισμός: Δυαδική επιλογή

OPT(j) = τιμή της βέλτιστης λύσης προβλήματος αποτελούμενο από αιτήματα 1, 2, ..., j

- Περίπτωση 1: Η OPT(j) επιλέγει το αίτημα j, με βαρύτητα v_j
  - Δεν μπορούν να χρησιμοποιηθούν τα ασύμβατα αιτήματα { p(j) + 1, p(j) + 2, ..., j - 1 }
  - Πρέπει να περιέχει την βέλτιστη λύση **OPT(p(j))** στο πρόβλημα που αποτελείται από τα συμβατά αιτήματα {1, 2, ..., p(j)} που έχουν απομείνει

Βέλτιστο υποπρόβλημα

- Περίπτωση 2: Η OPT(j) δεν επιλέγει το αίτημα j
  - Πρέπει να περιέχει την βέλτιστη λύση **OPT(j – 1)** στο πρόβλημα που αποτελείται από τα συμβατά αιτήματα {1, 2, ..., j-1} που έχουν απομείνει

$$OPT(j) = \begin{cases} 0 & \text{αν } j = 0 \\ \max\{v_j + OPT(p(j)), OPT(j-1)\} & \text{αλλιώς} \end{cases}$$

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων: Ωμή Βία

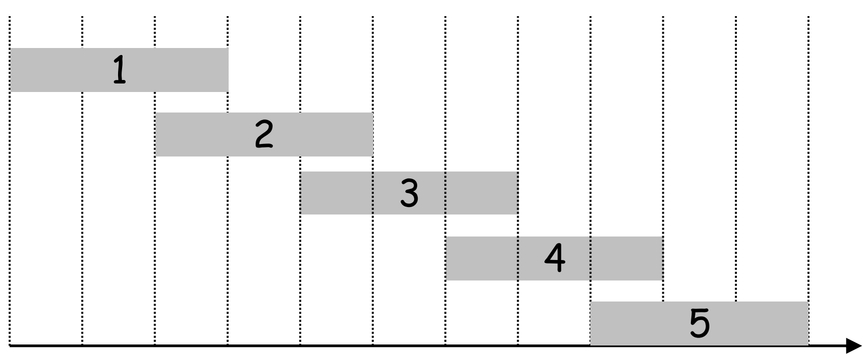
Αλγόριθμος ωμής βίας.

```
Είσοδος: n, s1,…,sn , f1,…,fn , v1,…,vn

Ταξινόμησε τα αιτήματα κατά χρόνο λήξης f1 ≤ f2 ≤ ... ≤ fn.

Υπολόγισε p(1), p(2), …, p(n)
Υπολόγισε Compute-Opt(1), Compute-Opt(2),…, Compute-Opt(n)

Compute-Opt(j) {
   if (j = 0)
      return 0
   else
      return max{vj + Compute-Opt(p(j)), Compute-Opt(j-1)}
}
```

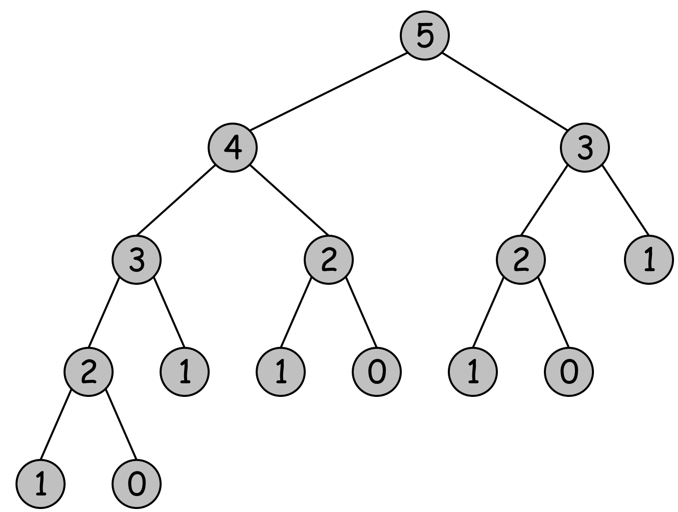
Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων: Ωμή Βία

Παρατήρηση. Ο αναδρομικός αλγόριθμος αποτυγχάνει θεαματικά λόγω επανϋπολογισμού πλεοναζόντων υποπροβλημάτων ⇒ εκθετικός αλγόριθμος

Παράδειγμα. Ο αριθμός των αναδρομικών κλήσεων στην παρακάτω περίπτωση μεγαλώνει σαν μια ακολουθία Fibonacci T(n) = T(n-1) + T(n-2) (δηλαδή εκθετική)



p(1) = 0, p(j) = j-2



Compute-Opt(j): παράγει αναδρομικές κλήσεις σε υποπροβλήματα μεγέθους j-1 και j-2

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων

- **Παρατήρηση 1.** Ο Compute-Opt επιλύει n+1 διαφορετικά υποπροβλήματα Compute-Opt(0), Compute-Opt(1), …., Compute-Opt(n)

- **Παρατήρηση 2.** Εκθετική πολυπλοκότητα του Compute-Opt οφείλεται στον εντυπωσιακό πλεονασμό κλήσεων κάθε ενός από τα Compute-Opt(j)

- *Ερώτημα. Πως απαλείφουμε αυτόν τον πλεονασμό;*

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων – Απομνημόνευση

Απομνημόνευση.

- Αποθήκευσε τα αποτελέσματα κάθε υποπροβλήματος σε μια καθολικά προσπελάσιμη θέση μνήμης.
- Αναζήτηση όποτε χρειαστεί.

```
Είσοδος: n, s1,…,sn , f1,…,fn , v1,…,vn

Ταξινόμησε τα αιτήματα κατά χρόνο λήξης, έτσι ώστε f1 ≤ f2 ≤ ... ≤ fn
Υπολόγισε p(1), p(2), …, p(n)

for j = 1 to n
   M[j] = empty     ← Καθολικός πίνακας απομνημόνευσης
M[0] = 0
M-Compute-Opt(n)

M-Compute-Opt(j) {
   if (M[j] is empty)
      M[j] = max{vj + M-Compute-Opt(p(j)), M-Compute-Opt(j-1)}
   return M[j]
}
```

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων – Χρόνος Εκτέλεσης

Ισχυρισμός. Η εκδοχή με απομνημόνευση του αλγόριθμου παίρνει χρόνο $O(n \log n)$

Απόδειξη.

- Ταξινόμηση ως προς χρόνο λήξης: $O(n \log n)$.
- Υπολογισμός $p(\cdot)$: $O(n)$ μετά από ταξινόμηση ως προς τον χρόνο έναρξης.

- `M-Compute-Opt(j)`: κάθε κλήση απαιτεί χρόνο $O(1)$ και είτε
  - (i) επιστρέφει μια υπάρχουσα τιμή `M[j]`
  - (ii) συμπληρώνει μια νέα εγγραφή `M[j]` και εκτελεί δύο αναδρομικές κλήσεις

- Μέτρο προόδου Φ = # μη κενών εγγραφών του `M[]`.
  - Αρχικά $\Phi = 0$, κατά τη διάρκεια $\Phi \leq n$.
  - Το (ii) αυξάνει το Φ κατά 1 $\Rightarrow$ το πολύ $2n$ αναδρομικές κλήσεις.
- Ο συνολικός χρόνος εκτέλεσης του `M-Compute-Opt(n)` είναι $O(n)$. ▪

Παρατήρηση. $O(n)$ αν τα αιτήματα είναι ταξινομημένα εξ'αρχής κατά χρόνους έναρξης και λήξης.

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων - Εύρεση Λύσης

Ερ. Ο αλγόριθμος βρίσκει τη βέλτιστη τιμή. Τι γίνεται αν θέλουμε την ίδια την λύση ;

Απ. Εκτελούμε ένα επιπλέον βήμα μετεπεξεργασίας του M, «ινχηλατώντας» τη λύση

```
Run M-Compute-Opt(n)
Run Find-Solution(n)

Find-Solution(j) {
   if (j = 0)
      μην εκτυπώσεις τίποτα
   else if (v_j + M[p(j)] > M[j-1])
      εκτύπωσε το j
      Find-Solution(p(j))
   else
      Find-Solution(j-1)
}
```

- \# επαναληπτικών κλήσεων $\leq n \Rightarrow O(n)$.

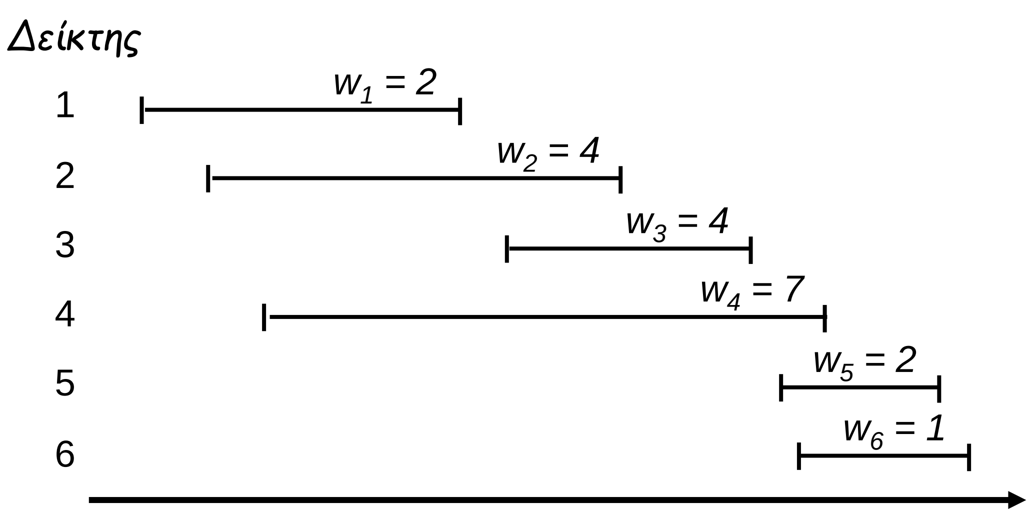
Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

Βασική Ιδέα. «Ξεδίπλωσε» την αναδρομή.

```
Είσοδος: n, s1,…,sn , f1,…,fn , v1,…,vn

Ταξινόμησε αιτήματα ως προς χρόνους λήξης f1 ≤ f2 ≤ ... ≤ fn

Υπολόγισε p(1), p(2), …, p(n)

Iterative-Compute-Opt {
   M[0] = 0
   for j = 1 to n
      M[j] = max{vj + M[p(j)], M[j-1]}
}
```

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

```
Είσοδος: n, s_1,…,s_n , f_1,…,f_n , v_1,…,v_n
Ταξινόμησε αιτήματα ως προς χρόνους λήξης f_1 ≤ f_2 ≤ ... ≤ f_n
Υπολόγισε p(1), p(2), …, p(n)
Iterative-Compute-Opt {
   M[0] = 0
   for j = 1 to n
      M[j] = max{w_j + M[p(j)], M[j-1]}
}
```



$p(1) = 0$

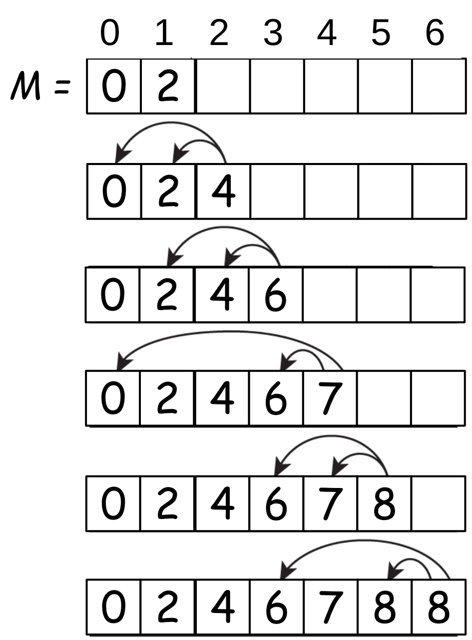
$p(2) = 0$

$p(3) = 1$

$p(4) = 0$

$p(5) = 3$

$p(6) = 3$



Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

$$M[j] = \max(w_j + M[p(j)],\ M[j-1])$$

$M[0] = 0$

$M[1] = \max(w_1 + M[0],\ M[0]) = \max(2 + 0,\ 0) = 2$

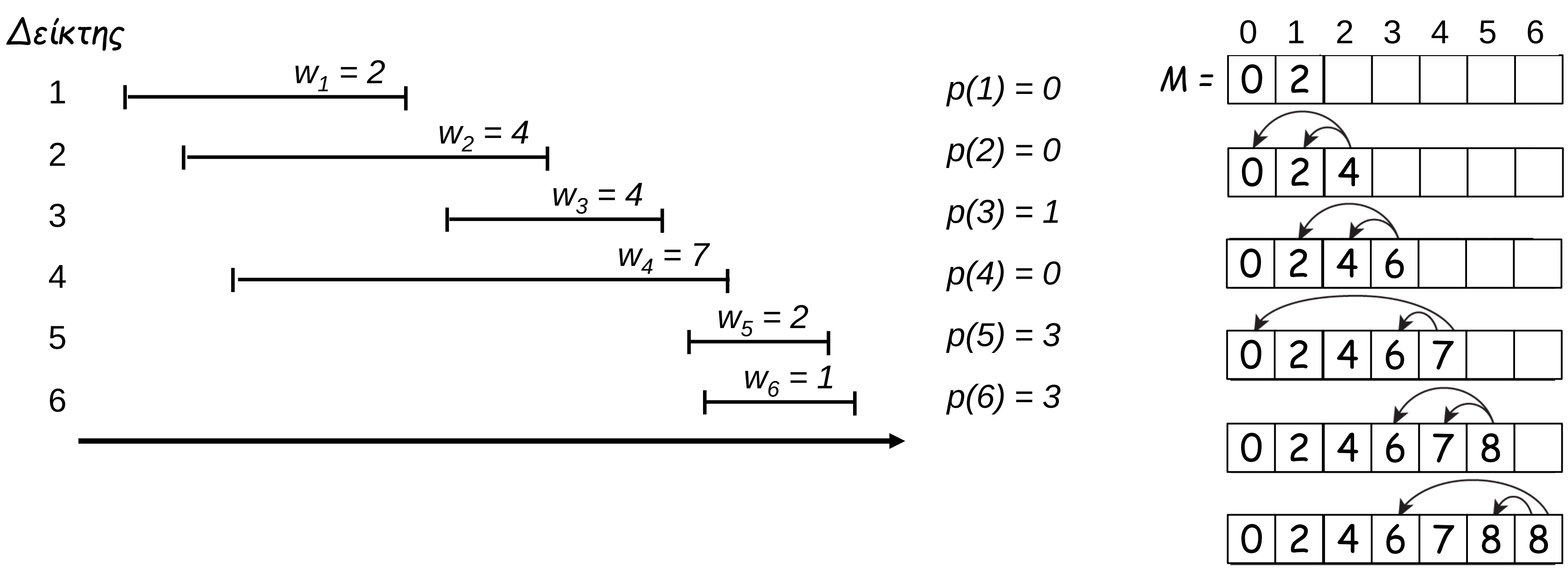
$M[2] = \max(w_2 + M[0],\ M[1]) = \max(4 + 0,\ 2) = 4$

$M[3] = \max(w_3 + M[1],\ M[2]) = \max(4 + 2,\ 4) = 6$

$M[4] = \max(w_4 + M[0],\ M[3]) = \max(7 + 0,\ 6) = 7$

$M[5] = \max(w_5 + M[3],\ M[4]) = \max(2 + 6,\ 7) = 8$

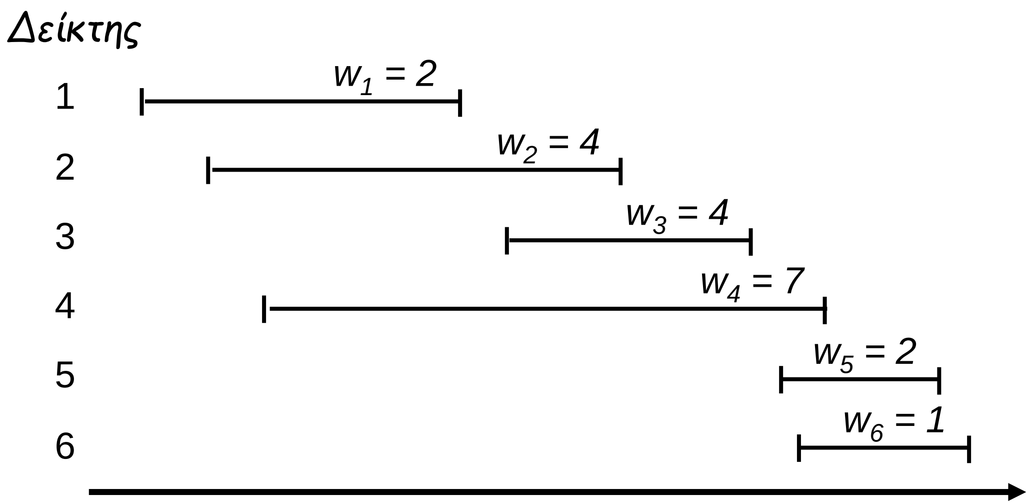
$M[6] = \max(w_6 + M[3],\ M[5]) = \max(1 + 6,\ 8) = 8$



Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

$$M[j] = \max(w_j + M[p(j)],\ M[j-1])$$

$$
\begin{aligned}
M[0] &= 0\\
M[1] &= \max(w_1 + M[0],\ M[0]) = \max(2 + 0,\ 0) = 2\\
M[2] &= \max(w_2 + M[0],\ M[1]) = \max(4 + 0,\ 2) = 4\\
M[3] &= \max(w_3 + M[1],\ M[2]) = \max(4 + 2,\ 4) = 6\\
M[4] &= \max(w_4 + M[0],\ M[3]) = \max(7 + 0,\ 6) = 7\\
M[5] &= \max(w_5 + M[3],\ M[4]) = \max(2 + 6,\ 7) = 8\\
M[6] &= \max(w_6 + M[3],\ M[5]) = \max(1 + 6,\ 8) = 8
\end{aligned}
$$



$p(1) = 0$

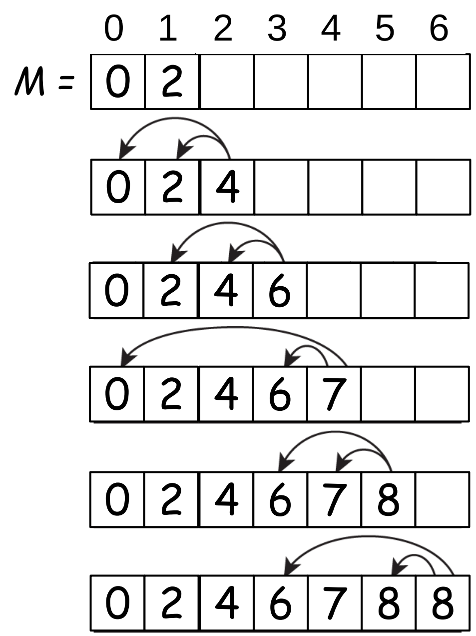
$p(2) = 0$

$p(3) = 1$

$p(4) = 0$

$p(5) = 3$

$p(6) = 3$



Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

$$M[j] = \max(w_j + M[p(j)],\ M[j-1])$$

$M[0] = 0$

$M[1] = \max(w_1 + M[0],\ M[0]) = \max(2 + 0,\ 0) = 2$

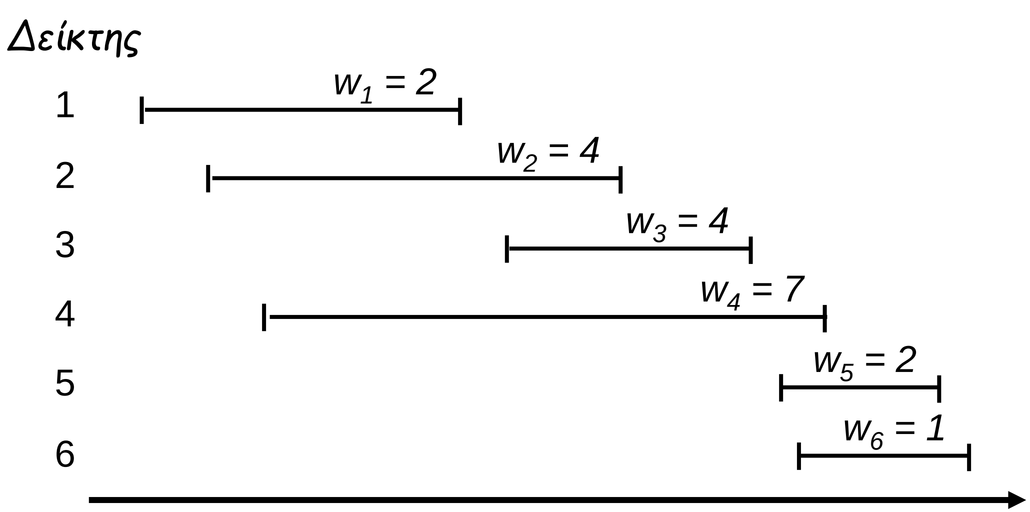
$M[2] = \max(w_2 + M[0],\ M[1]) = \max(4 + 0,\ 2) = 4$

$M[3] = \max(w_3 + M[1],\ M[2]) = \max(4 + 2,\ 4) = 6$

$M[4] = \max(w_4 + M[0],\ M[3]) = \max(7 + 0,\ 6) = 7$

$M[5] = \max(w_5 + M[3],\ M[4]) = \max(2 + 6,\ 7) = 8$ ⇒ εργασία 5

$M[6] = \max(w_6 + M[3],\ M[5]) = \max(1 + 6,\ 8) = 8$



$p(1) = 0$

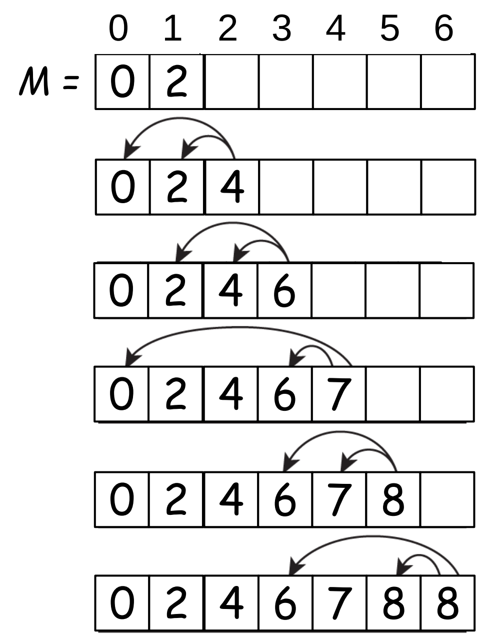
$p(2) = 0$

$p(3) = 1$

$p(4) = 0$

$p(5) = 3$

$p(6) = 3$



Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

$$M[j] = \max(w_j + M[p(j)],\ M[j-1])$$

$M[0] = 0$

$M[1] = \max(w_1 + M[0],\ M[0]) = \max(2 + 0,\ 0) = 2$

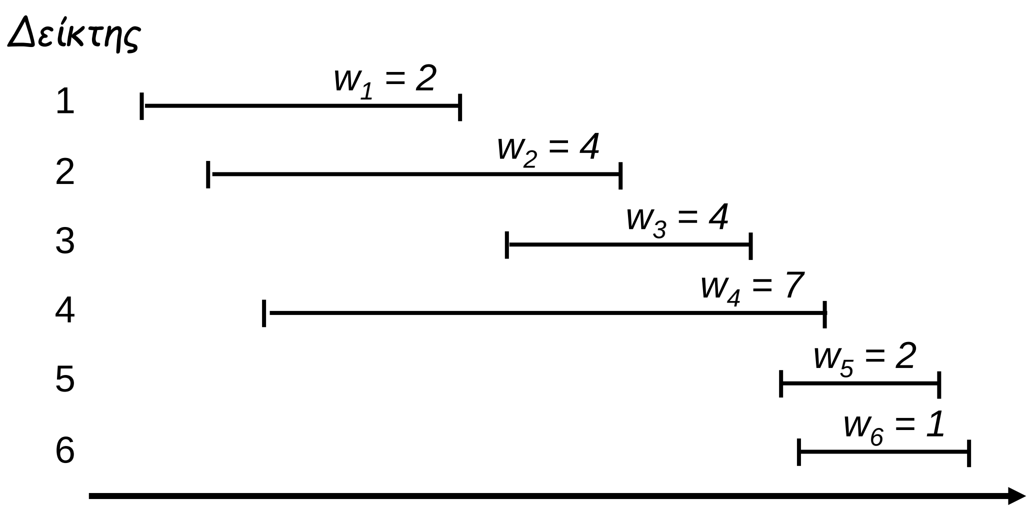
$M[2] = \max(w_2 + M[0],\ M[1]) = \max(4 + 0,\ 2) = 4$

$M[3] = \max(w_3 + M[1],\ M[2]) = \max(4 + 2,\ 4) = 6$

$M[4] = \max(w_4 + M[0],\ M[3]) = \max(7 + 0,\ 6) = 7$

$M[5] = \max(w_5 + M[3],\ M[4]) = \max(2 + 6,\ 7) = 8$ ⇒ εργασία 5

$M[6] = \max(w_6 + M[3],\ M[5]) = \max(1 + 6,\ 8) = 8$



$p(1) = 0$

$p(2) = 0$

$p(3) = 1$

$p(4) = 0$

$p(5) = 3$

$p(6) = 3$

	0	1	2	3	4	5	6
M =	0	2					
	0	2	4				
	0	2	4	6			
	0	2	4	6	7		
	0	2	4	6	7	8	
	0	2	4	6	7	8	8

Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

$$M[j] = \max(w_j + M[p(j)],\ M[j-1])$$

$M[0] = 0$

$M[1] = \max(w_1 + M[0],\ M[0]) = \max(2 + 0,\ 0) = 2$

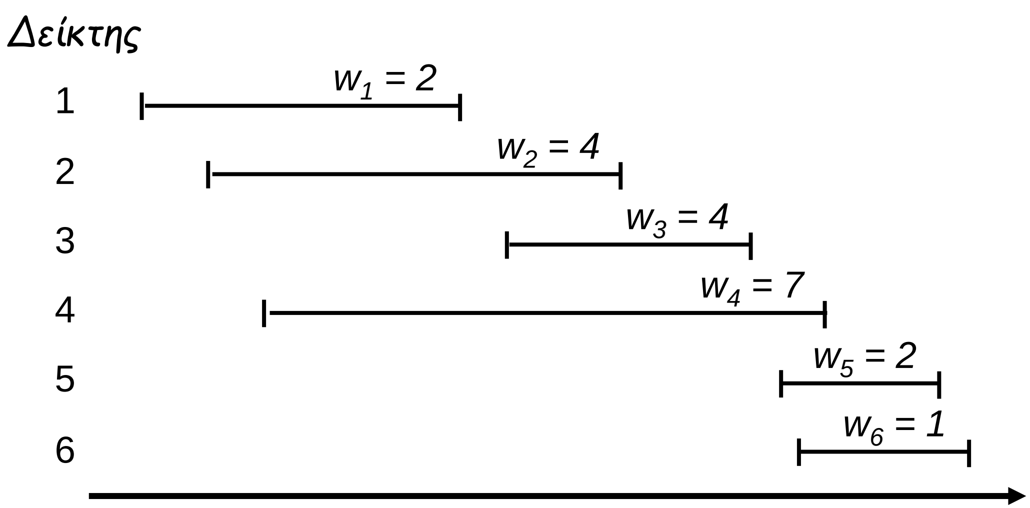
$M[2] = \max(w_2 + M[0],\ M[1]) = \max(4 + 0,\ 2) = 4$

$M[3] = \max(w_3 + M[1],\ M[2]) = \max(4 + 2,\ 4) = 6$ ⇒ εργασία 3

$M[4] = \max(w_4 + M[0],\ M[3]) = \max(7 + 0,\ 6) = 7$

$M[5] = \max(w_5 + M[3],\ M[4]) = \max(2 + 6,\ 7) = 8$ ⇒ εργασία 5

$M[6] = \max(w_6 + M[3],\ M[5]) = \max(1 + 6,\ 8) = 8$



$p(1) = 0$

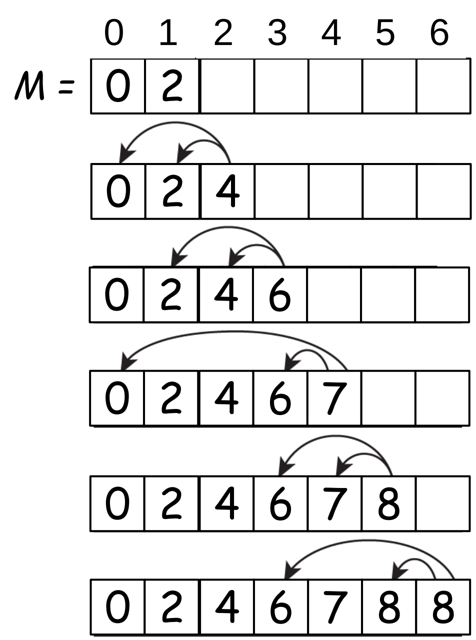
$p(2) = 0$

$p(3) = 1$

$p(4) = 0$

$p(5) = 3$

$p(6) = 3$



Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

$$M[j] = \max(w_j + M[p(j)],\ M[j-1])$$

$M[0] = 0$

$M[1] = \max(w_1 + M[0],\ M[0]) = \max(2 + 0,\ 0) = 2$

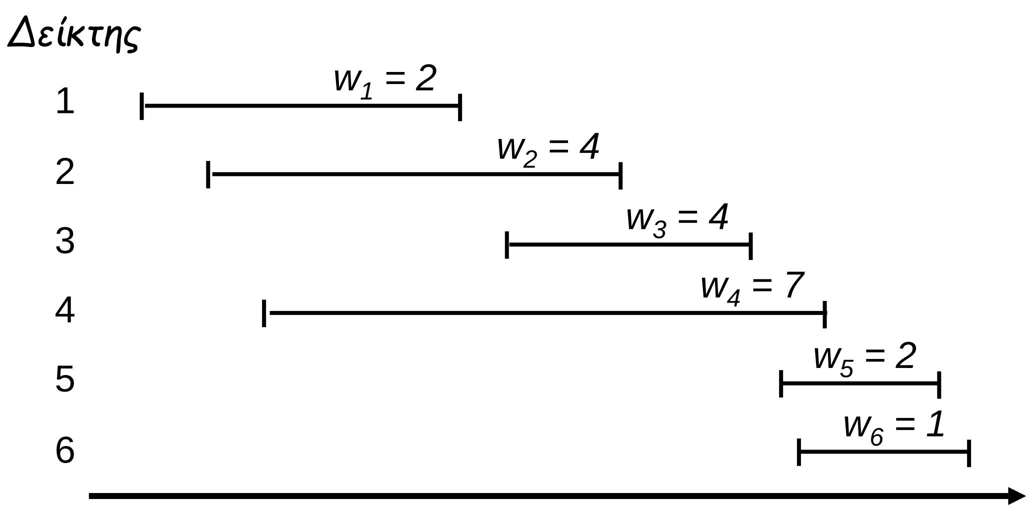
$M[2] = \max(w_2 + M[0],\ M[1]) = \max(4 + 0,\ 2) = 4$

$M[3] = \max(w_3 + M[1],\ M[2]) = \max(4 + 2,\ 4) = 6$ ⇒ εργασία 3

$M[4] = \max(w_4 + M[0],\ M[3]) = \max(7 + 0,\ 6) = 7$

$M[5] = \max(w_5 + M[3],\ M[4]) = \max(2 + 6,\ 7) = 8$ ⇒ εργασία 5

$M[6] = \max(w_6 + M[3],\ M[5]) = \max(1 + 6,\ 8) = 8$



$p(1) = 0$

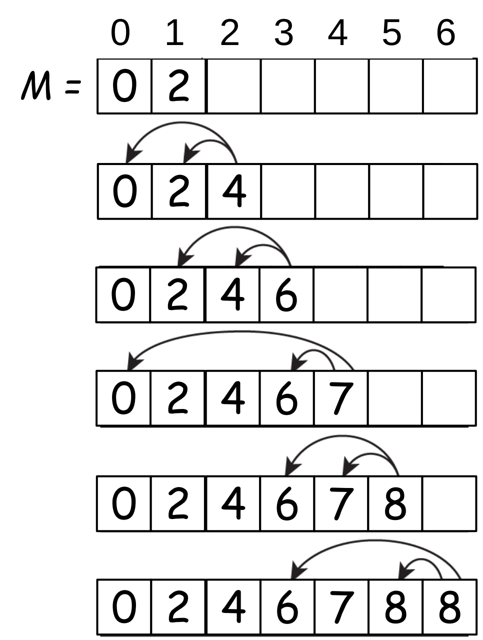
$p(2) = 0$

$p(3) = 1$

$p(4) = 0$

$p(5) = 3$

$p(6) = 3$



Σταθμισμένος Χρονοπρογραμματισμός Διαστημάτων
Απομνημόνευση ή επανάληψη στα υποπροβλήματα

$$M[j] = \max(w_j + M[p(j)],\ M[j-1])$$

$M[0] = 0$

$M[1] = \max(w_1 + M[0],\ M[0]) = \max(2 + 0,\ 0) = 2$ ⇒ εργασία 1

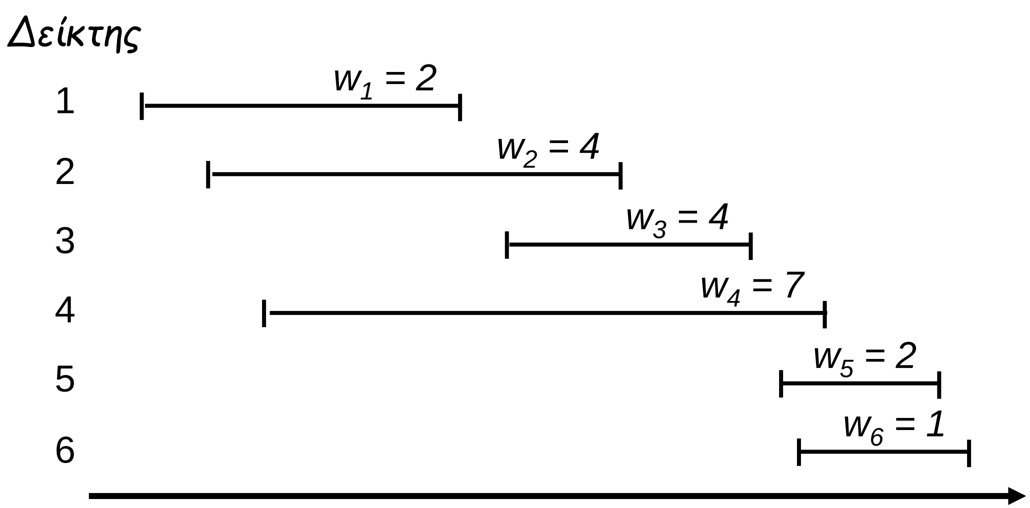
$M[2] = \max(w_2 + M[0],\ M[1]) = \max(4 + 0,\ 2) = 4$

$M[3] = \max(w_3 + M[1],\ M[2]) = \max(4 + 2,\ 4) = 6$ ⇒ εργασία 3

$M[4] = \max(w_4 + M[0],\ M[3]) = \max(7 + 0,\ 6) = 7$

$M[5] = \max(w_5 + M[3],\ M[4]) = \max(2 + 6,\ 7) = 8$ ⇒ εργασία 5

$M[6] = \max(w_6 + M[3],\ M[5]) = \max(1 + 6,\ 8) = 8$



$p(1) = 0$

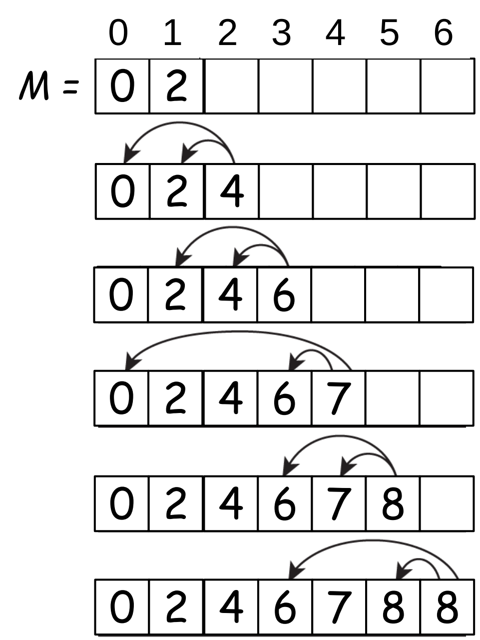
$p(2) = 0$

$p(3) = 1$

$p(4) = 0$

$p(5) = 3$

$p(6) = 3$



Σύνοψη Δυναμικού Προγραμματισμού

Συνταγή.

- Χαρακτηρισμός δομής του προβλήματος.
- Αναδρομικά ορισμός της τιμής της βέλτιστης λύσης.
- Υπολογισμός της τιμής της βέλτιστης λύσης.
- Κατασκευή της βέλτιστης λύσης από την υπολογισμένη πληροφορία.

Βιβλιογραφία

1. J. Kleinberg and E. Tardos, *Σχεδιασμός Αλγορίθμων*, ελληνική έκδοση, Εκδόσεις Κλειδάριθμος, 2008
2. T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Εισαγωγή στους Αλγορίθμους*, ελληνική έκδοση, Πανεπιστημιακές Εκδόσεις Κρήτης, 2012
3. K. Mehlhorn and P. Sanders, *Αλγόριθμοι και Δομές Δεδομένων – Τα βασικά εργαλεία*, ελληνική έκδοση, Εκδόσεις Κλειδάριθμος, 2014
4. S. Dasgupta, C. Papadimitriou, and U. Vazirani, *Αλγόριθμοι*, ελληνική έκδοση, Εκδόσεις Κλειδάριθμος, 2008
5. Θ. Παπαθεοδώρου, *Αλγόριθμοι: Εισαγωγικά Θέματα και Παραδείγματα*, Εκδόσεις Πανεπιστημίου Πατρών, 1999

Τέλος Ενότητας

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.

Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Χρήστος Ζαρολιάγκης, 2014. «Εισαγωγή στους Αλγορίθμους». Έκδοση: 1.0. Πάτρα 2014. Διαθέσιμο από τη δικτυακή διεύθυνση:

https://eclass.upatras.gr/courses/CEID1083

Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση, Όχι Παράγωγα Έργα 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό.

[1] http://creativecommons.org/licenses/by-nc-nd/4.0/

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.