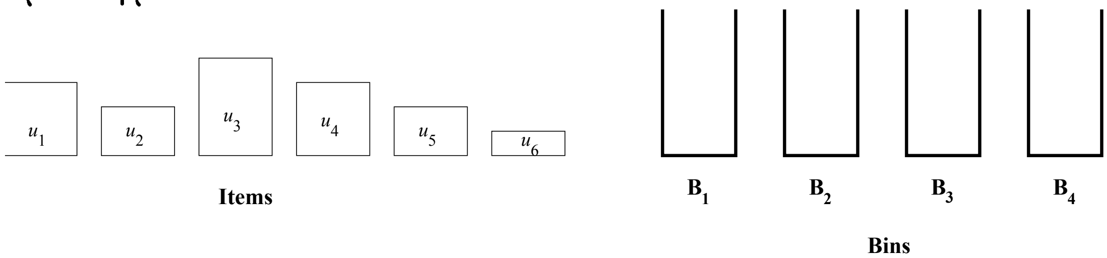
Bin-Packing

Πρόβλημα Δίνεται πεπερασμένο σύνολο $U = \{u_1, u_2, \ldots, u_n\}$ από *αντικείμενα* και μια *συνάρτηση μεγέθους* $s(u) \in [0 \ldots 1]$ για κάθε αντικείμενο $u \in U$. Ζητείται μια διαμέριση του U σε ξένα μεταξύ τους σύνολα $U_1, U_2, \ldots, U_k$ έτσι ώστε το άθροισμα των μεγεθών των αντικειμένω σε κάθε κάδο (bin) U_i να είναι το πολύ 1, και ο αριθμός των κάδων k να είναι όσο το δυνατόν μικρότερος.

- Θεωρούμε κάθε U_i ως *κάδο χωρητικότητας μιας μονάδας*.

Παράδειγμα



- Έστω $OPT(I)$ το πλήθος των κάδων σε μία βέλτιστη λύση του προβλήματος bin-packing για το στιγμιότυπο I. Τότε,

$$OPT(I) \geq \left\lceil \sum_{i=1}^{n} s(u_i) \right\rceil$$

- **First-Fit approach**

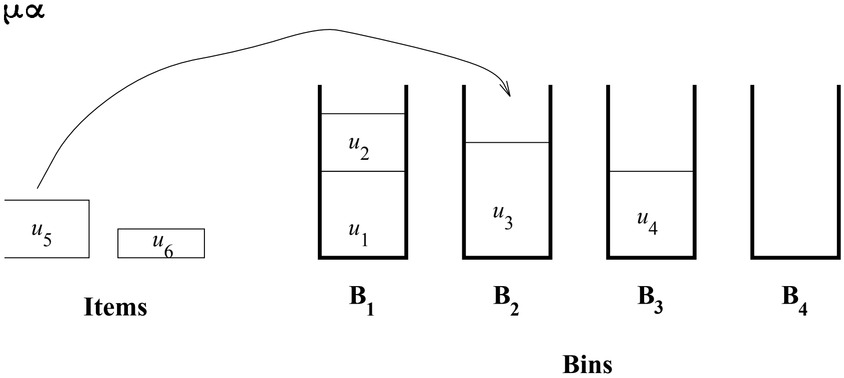
First-Fit(I)

1. Start with an infinite sequence $B_1,\ B_2, \ldots$ of unit-capacity bins, all of which are empty.

2. For $i = 1$ **to** n **do**

Place item u_i into the bin of smallest index in which it can fit.

Παράδειγμα



Έστω $FF(I)$ το πλήθος των κάδων που χρειάζονται για τον αλγόριθμο *First-Fit*() για το στιγμιότυπο I.

Θεώρημα $FF(I) \leq 2 \cdot OPT(I)$

Απόδειξη

- Έχουμε ότι $FF(I) < \left\lceil 2 \sum_{i=1}^{n} s(u_i) \right\rceil$ (1)
  εφόσον μπορεί να υπάρχει το πολύ 1 κάδος που δεν είναι άδειος και έχει συνολικό περιεχόμενο $\frac{1}{2}$ ή λιγότερο.

- Επιπλέον έχουμε ότι $OPT(I) \geq \left\lceil \sum_{i=1}^{n} s(u_i) \right\rceil$ (2)

- $(1), (2) \implies FF(I) \leq 2 \cdot OPT(I)$

□

Θεώρημα i) $FF(I) \leq \frac{17}{10} \cdot OPT(I) + 2$, για κάθε στιγμιότυπο I.

ii) Υπάρχουν στιγμιότυπα όπου το $OPT(I)$ μπορεί να είναι αυθαίρετα μεγάλο και να ισχύει
$FF(I) \geq \frac{17}{10}(OPT(I) - 1)$

□

- **First-Fit-Decreasing approach**

First-Fit-Decreasing(I)

1. Sort the items in decreasing order of size.

2. Apply *First-Fit*() in the sequence of items obtained from step 1.

- Έστω $FFD(I)$ το πλήθος των κάδων που απαιτούνται από τον αλγόριθμο *First-Fit-Decreasing*() για το στιγμιότυπο I.

Θεώρημα i) $FFD(I) \leq \frac{11}{9} \cdot OPT(I) + 4$, για κάθε στιγμιότυπο I.

ii) Υπάρχουν στιγμιότυπα όπου το $OPT(I)$ μπορεί να είναι αυθαίρετα μεγάλο και να ισχύει
$FFD(I) = \frac{11}{9} \cdot OPT(I)$

□

Αλγόριθμοι Ψευδο-Πολυωνυμικού χρόνου

- **Το πρόβλημα "Partition"**

Είσοδος: Πεπερασμένο σύνολο A και *βάρη* $s(a) \in Z^+$ για κάθε $a \in A$.

Έξοδος: Υπάρχει υποσύνολο $A' \subset A$ τέτοιο ώστε $\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)$;

Παράδειγμα

$A = \{a_1, a_2, a_3, a_4, a_5\}$

$s(a_1) = 1$
$s(a_2) = 9$
$s(a_3) = 5$
$s(a_4) = 3$
$s(a_5) = 8$

Έστω $A' = \{a_3, a_5\}$

$\Longrightarrow \quad \sum_{a \in A'} s(a) = a_3 + a_5 = 5 + 8 = 13$

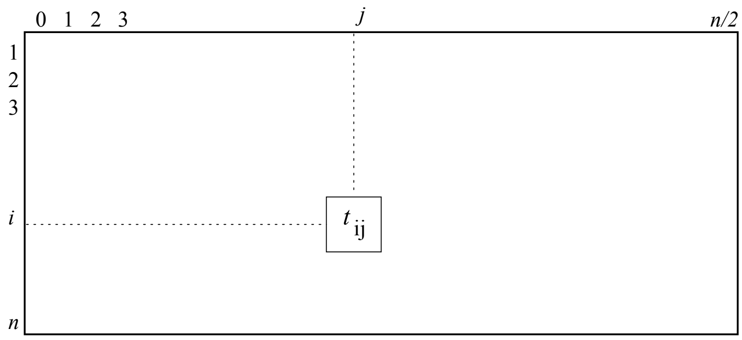
$\Longrightarrow \quad \sum_{a \in A - A'} s(a) = a_1 + a_2 + a_4 = 1 + 9 + 3 = 13$

Έστω $B = \sum_{a \in A} s(a)$

- Αν το B είναι περιττό, προφανώς κανένα σύνολο A' δεν ικανοποιεί τις προϋποθέσεις του προβλήματος.
- Για $1 \leq i \leq n, \; 0 \leq j \leq \frac{B}{2}$ θεωρούμε την πρόταση:

  "Υπάρχει υποσύνολο του $\{a_1, a_2, \ldots, a_i\}$ τέτοιο ώστε το άθροισμα των βαρών των αντικειμένων του είναι ακριβώς j."

  Έστω $t(i, j)$ η αληθοτιμή της πρότασης αυτής.
- Οι τιμές των $t(i, j)$ μπορούν να τοποθετηθούν σε πίνακα διαστάσεων $n \times (\frac{B}{2} + 1)$.



(Στο παράδειγμά μας, $n = 5$ και $B = \frac{26}{2} = 13$.)

- Μπορούμε να συμπληρώσουμε τον πίνακα ως εξής:

Γραμμή 1 $t(1,j) = \mathbf{T} \iff j = 0$ ή $j = s(a_1)$

Γραμμή $\boldsymbol{i}$ $t(i,j) = \mathbf{T} \iff t(i-1,j) = \mathbf{T}$ ή $(s(a_i) \leq j$ και $t(i-1, j - s(a_i)) = \mathbf{T})$

Στο παράδειγμά μας, $s(a_1) = 1$, $s(a_2) = 9$, $s(a_3) = 5$, $s(a_4) = 3$, $s(a_5) = 8$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**T**	**T**												
2														
3														
4														
5														

Πολυπλοκότητα $O(nB)$

Γεγονός: Το πρόβλημα *"partition"* είναι NP-complete.

Ερώτηση Αποδείξαμε ότι $P = NP$;

Απάντηση ΌΧΙ!

- **ΔΕΝ** έχουμε πολυωνυμικό αλγόριθμο ως προς **το μέγεθος της εισόδου**.

- Κάθε ακέραιος $s(a_i)$ μπορεί να γραφεί ως μια ακολουθία ψηφίων μήκους $O(\log s(a_i))$.
  Συνεπώς, το μέγεθος της εισόδου I έχει μήκος

$$\mathit{length}(I) = \sum_{i=1}^{n} \log s(a_i) \leq \sum_{i=1}^{n} \log B = O(n \log B)$$

- Το nB δεν φράσεται από πολυωνυμική συνάρτηση του $n \log B$.

• Οι Ψευδο-Πολυωνυμικοί αλγόριθμοι είναι χρήσιμοι στην πράξη.

  • Σε προβλήματα *scheduling*, οι αριθμοί αντιπροσωπεύουν την διάρκεια μιας εργασίας και είναι λογικό να θεωρήσουμε ότι είναι μικροί.
  • Σε προβλήματα όπου *οι αριθμοί αντιπροσωπεύουν ποσότητες που εκτιμώνται εμπειρικά*, οι περιορισμοί στην ακρίβρεια την μέτρησης περιορίζουν το εύρος τους.
  • Σε εφαρμογές όπου δεν τίθενται περιορισμοί σε αριθμούς, οι ψευδο-πολυωνυμικοί αλγόριθμοι εμφανίζουν *"εκθετική συμπεριφορά"* μόνο όταν αντιμετωπίζουν στιγμιότυπα που περιέχουν *"εκθετικά μεγάλους"* αριθμούς.

Reading Material

"Computers and Intractability. A Guide to the Theory of NP-Completeness", Michael R. Garey and David S. Johnson, W.H. Freeman and Company, 1979.

"Bin-packing" pp. 123–128.

"Pseudo-polynomial algorithms" pp. 90–92.

Τελειώνοντας, θυμηθείτε:

• Υπάρχουν

LIES,

BIG LIES, και

BIG "O" NOTATION.

• ΠΑΝΤΑ να υπολογίζετε την κρυμένη σταθερά στην ασυμπτωτική πολυπλοκότητα ενός αλγορίθμου που εφαρμόζετε.

• ΠΑΝΤΑ να υπολογίζετε τις ιδιότητες που έχει η είσοδός σας.