Αλγόριθμοι και Πολυπλοκότητα

Αρχοντία Γιαννοπούλου
Όλγα Φουρτουνέλλη

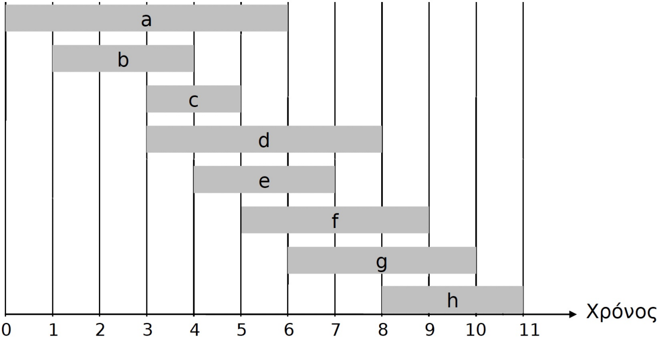
Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών

Άπληστοι Αλγόριθμοι

Χρονοπρογραμματισμός Διαστημάτων (Scheduling)

Χρονοπρογραμματισμός Διαστημάτων (Scheduling)

- Η εργασία j ξεκινάει την χρονική στιγμή s_j και τελειώνει την στιγμή f_j.
- Δύο εργασίες είναι συμβατές αν δεν επικαλύπτονται χρονικά.
- Σκοπός (βελτιστοποίηση): βρείτε το μεγαλύτερο υποσύνολο (σε πλήθος) από συμβατές μεταξύ τους εργασίες.



Χρονοπρογραμματισμός Διαστημάτων

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί.

Χρονοπρογραμματισμός Διαστημάτων

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί.

- Μικρότερος χρόνος έναρξης: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς τον χρόνο έναρξης s_j.

Χρονοπρογραμματισμός Διαστημάτων

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί.

- Μικρότερος χρόνος έναρξης: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς τον χρόνο έναρξης s_j.

- Μικρότερος χρόνος λήξης: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς τον χρόνο λήξης f_j.

Χρονοπρογραμματισμός Διαστημάτων

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί.

- Μικρότερος χρόνος έναρξης: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς τον χρόνο έναρξης s_j.

- Μικρότερος χρόνος λήξης: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς τον χρόνο λήξης f_j.

- Μικρότερο χρονικό διάστημα: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς το χρονικό διάστημα $f_j - s_j$.

Χρονοπρογραμματισμός Διαστημάτων

Άπληστο πρότυπο. Θεωρήστε τις εργασίες σε κάποια σειρά. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν ήδη επιλεγεί.

- Μικρότερος χρόνος έναρξης: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς τον χρόνο έναρξης s_j.
- Μικρότερος χρόνος λήξης: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς τον χρόνο λήξης f_j.
- Μικρότερο χρονικό διάστημα: Θεωρήστε τις εργασίες σε αύξουσα σειρά ως προς το χρονικό διάστημα $f_j - s_j$.
- Ελάχιστες διενέξεις: Για κάθε εργασία, μετρήστε τον αριθμό των μη συμβατών εργασιών c_j. Προγραμματίστε σε αύξουσα σειρά των διενέξεων (συγκρούσεων) c_j.

Ο μικρότερος χρόνος έναρξης δεν αποδίδει

Η εργασία που ξεκινάει πρώτη μπορεί να έχει πολλές επικαλύψεις και να μας εμποδίζει από το να επιλέξουμε μεγαλύτερο πλήθος εργασιών.

Το μικρότερο χρονικό διάστημα δεν αποδίδει

Η εργασία με τη μικρότερη δυνατή διάρκεια μπορεί να έχει επικάλυψη και να μας εμποδίζει από το να επιλέξουμε μεγαλύτερο πλήθος εργασιών.

Οι ελάχιστες διενέξεις δεν αποδίδουν

Η εργασία με τη μικρότερη δυνατή διάρκεια μπορεί να έχει επικάλυψη παρόμοια με το προηγούμενο παράδειγμα και να μας εμποδίζει από το να επιλέξουμε μεγαλύτερο πλήθος εργασιών.

Άπληστος αλγόριθμος

Θεωρήστε τις εργασίες σε αύξουσα σειρά χρόνων λήξης. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν επιλεγεί.

Άπληστος αλγόριθμος

Θεωρήστε τις εργασίες σε αύξουσα σειρά χρόνων λήξης. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν επιλεγεί.

Ταξινόμησε τις εργασίες κατά το χρόνο λήξης $f_1 \leq f_2 \leq \cdots \leq f_n$
$A \leftarrow \emptyset$
Για $j = 1$ έως n
  Εάν η εργασία j είναι συμβατή με το A
    $A \leftarrow A \cup \{j\}$.
Επίστρεψε A.

Άπληστος αλγόριθμος

Θεωρήστε τις εργασίες σε αύξουσα σειρά χρόνων λήξης. Επιλέξτε κάθε εργασία εφόσον είναι συμβατή με αυτές που έχουν επιλεγεί.

Ταξινόμησε τις εργασίες κατά το χρόνο λήξης $f_1 \leq f_2 \leq \cdots \leq f_n$
$A \leftarrow \emptyset$
Για $j = 1$ έως n
    Εάν η εργασία j είναι συμβατή με το A
        $A \leftarrow A \cup \{j\}$.
Επίστρεψε A.

Υλοποίηση: $\mathcal{O}(n \log n)$.

- Θυμηθείτε την εργασία j^* που προστέθηκε τελευταία στο A.
- Η εργασία j είναι συμβατή με το A αν $s_j \geq f_{j^*}$.

Ανάλυση

Θεώρημα

Ο άπληστος αλγόριθμος ως προς τον χρόνο λήξης είναι βέλτιστος.

Απόδειξη. (εις άτοπον)

Ανάλυση

Θεώρημα

Ο άπληστος αλγόριθμος ως προς τον χρόνο λήξης είναι βέλτιστος.

Απόδειξη. (εις άτοπον)

- Έστω ότι ο άπληστος αλγόριθμος δεν είναι βέλτιστος.

Ανάλυση

Θεώρημα

Ο άπληστος αλγόριθμος ως προς τον χρόνο λήξης είναι βέλτιστος.

Απόδειξη. (εις άτοπον)

- Έστω ότι ο άπληστος αλγόριθμος δεν είναι βέλτιστος.
- Έστω $i_1, \ldots, i_k$ το σύνολο εργασιών που επιλέγονται από τον άπληστο αλγόριθμο.

Ανάλυση

Θεώρημα

Ο άπληστος αλγόριθμος ως προς τον χρόνο λήξης είναι βέλτιστος.

Απόδειξη. (εις άτοπον)

- Έστω ότι ο άπληστος αλγόριθμος δεν είναι βέλτιστος.
- Έστω $i_1, \ldots, i_k$ το σύνολο εργασιών που επιλέγονται από τον άπληστο αλγόριθμο.
- Έστω $j_1, \ldots, j_m$ το σύνολο εργασιών στη βέλτιστη λύση με $i_1 = j_1, \ldots, i_r = j_r$ για τη μέγιστη δυνατή τιμή του r.

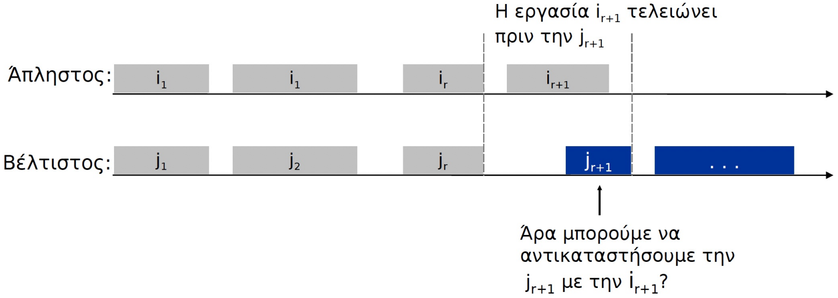
Ανάλυση

Θεώρημα

Ο άπληστος αλγόριθμος ως προς τον χρόνο λήξης είναι βέλτιστος.

Απόδειξη. (εις άτοπον)

- Έστω ότι ο άπληστος αλγόριθμος δεν είναι βέλτιστος.
- Έστω $i_1, \ldots, i_k$ το σύνολο εργασιών που επιλέγονται από τον άπληστο αλγόριθμο.
- Έστω $j_1, \ldots, j_m$ το σύνολο εργασιών στη βέλτιστη λύση με $i_1 = j_1, \ldots, i_r = j_r$ για τη μέγιστη δυνατή τιμή του r.



Διαμέριση Διαστημάτων

Διαμέριση Διαστημάτων

Η είσοδος μας είναι ένα σύνολο από n εργασίες όπου:

- Η εργασία j ξεκινάει τη χρονική στιγμή s_j και τελειώνει τη στιγμή f_j.

Στόχος: Η εύρεση του μικρότερου πλήθους μηχανών στις οποίες μπορούμε να αναθέσουμε τις εργασίες ώστε δύο εργασίες που τρέχουν ταυτόχρονα κάποια συγκεκριμένη χρονική στιγμή να μην ανατεθούν στην ίδια μηχανή.

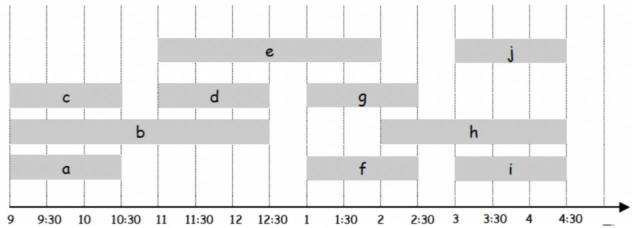
Διαμέριση Διαστημάτων

Η είσοδος μας είναι ένα σύνολο από n εργασίες όπου:

- Η εργασία j ξεκινάει τη χρονική στιγμή s_j και τελειώνει τη στιγμή f_j.

Στόχος: Η εύρεση του μικρότερου πλήθους μηχανών στις οποίες μπορούμε να αναθέσουμε τις εργασίες ώστε δύο εργασίες που τρέχουν ταυτόχρονα κάποια συγκεκριμένη χρονική στιγμή να μην ανατεθούν στην ίδια μηχανή.

Σε αυτή την ανάθεση χρειαζόμαστε 4 μηχανές για τις 10 εργασίες.



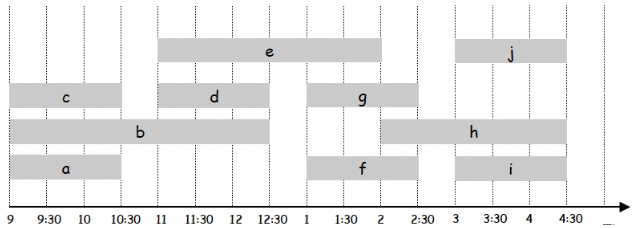
Διαμέριση Διαστημάτων

Η είσοδος μας είναι ένα σύνολο από n εργασίες όπου:

- Η εργασία j ξεκινάει τη χρονική στιγμή s_j και τελειώνει τη στιγμή f_j.

Στόχος: Η εύρεση του μικρότερου πλήθους μηχανών στις οποίες μπορούμε να αναθέσουμε τις εργασίες ώστε δύο εργασίες που τρέχουν ταυτόχρονα κάποια συγκεκριμένη χρονική στιγμή να μην ανατεθούν στην ίδια μηχανή.

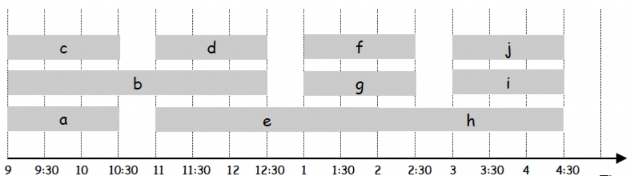
Σε αυτή την ανάθεση χρειαζόμαστε 4 μηχανές για τις 10 εργασίες.



Σε αυτή την ανάθεση χρειαζόμαστε 3 μηχανές για τις 10 εργασίες. Μπορούμε καλύτερα;

Καλύτερη λύση

Σε αυτή την ανάθεση χρειαζόμαστε 3 μηχανές για τις 10 εργασίες.



Μπορούμε καλύτερα;

Διαμέριση Διαστημάτων

Το βάθος ενός συνόλου ανοιχτών διαστημάτων είναι το μέγιστο πλήθος από αυτά που περιέχουν μία οποιαδήποτε δοσμένη χρονική στιγμή.

Διαμέριση Διαστημάτων

Το βάθος ενός συνόλου ανοιχτών διαστημάτων είναι το μέγιστο πλήθος από αυτά που περιέχουν μία οποιαδήποτε δοσμένη χρονική στιγμή.

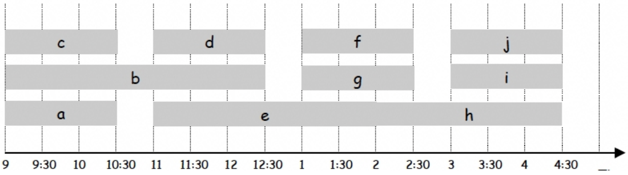
Χρειαζόμαστε τουλάχιστον τόσες μηχανές όσο το βάθος των διαστημάτων.

Διαμέριση Διαστημάτων

Το βάθος ενός συνόλου ανοιχτών διαστημάτων είναι το μέγιστο πλήθος από αυτά που περιέχουν μία οποιαδήποτε δοσμένη χρονική στιγμή.

Χρειαζόμαστε τουλάχιστον τόσες μηχανές όσο το βάθος των διαστημάτων.

Στο παράδειγμα το βάθος των διαστημάτων ισούται με 3 που είναι το μέγεθος της λύσης της εικόνας.

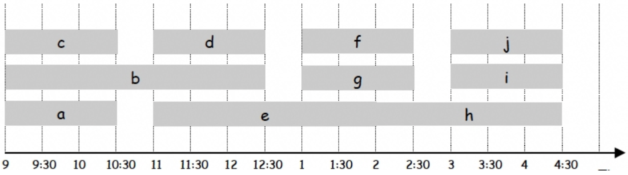


Διαμέριση Διαστημάτων

Το βάθος ενός συνόλου ανοιχτών διαστημάτων είναι το μέγιστο πλήθος από αυτά που περιέχουν μία οποιαδήποτε δοσμένη χρονική στιγμή.

Χρειαζόμαστε τουλάχιστον τόσες μηχανές όσο το βάθος των διαστημάτων.

Στο παράδειγμα το βάθος των διαστημάτων ισούται με 3 που είναι το μέγεθος της λύσης της εικόνας.



Υπάρχει πάντα χρονοπρογραμματισμός που το πλήθος των μηχανών να ισούται με το βάθος των διαστημάτων;

Διαμέριση Διαστημάτων - Αλγόριθμος

Άπληστο κριτήριο: Θεωρούμε τις εργασίες ταξινομημένες με βάση τη σειρά έναρξης. Αναθέτουμε τις εργασίες σε διαθέσιμες μηχαμές με βάση το χρόνο έναρξης.

Ταξινόμησε τις εργασίες κατά το χρόνο έναρξης $s_1 \leq s_2 \leq \cdots \leq s_n$.

$d \leftarrow 0$
Για $j = 1$ έως n
    Εάν η εργασία j είναι συμβατή με κάποια μηχανή k
        Αναθέτουμε την εργασία j στη μηχανή k
    Αλλιώς
        Δεσμεύουμε μία νέα μηχανή, έστω τη $d + 1$
        Αναθέτουμε την εργασία j στη μηχανή $d + 1$
        $d \leftarrow d + 1$

Επίστρεψε d

Ποιά είναι η χρονική πολυπλοκότητα του αλγόριθμου;

Διαμέριση Διαστημάτων - Ανάλυση Αλγόριθμου

Ο άπληστος αλγόριθμος που σχεδιάσαμε δεν αναθέτει ποτέ δύο συγκρουόμενες εργασίες στην ίδια μηχανή.

Διαμέριση Διαστημάτων - Ανάλυση Αλγόριθμου

Ο άπληστος αλγόριθμος που σχεδιάσαμε δεν αναθέτει ποτέ δύο συγκρουόμενες εργασίες στην ίδια μηχανή.

Θεώρημα: Ο άπληστος αλγόριθμος είναι βέλτιστος.

Διαμέριση Διαστημάτων - Ανάλυση Αλγόριθμου

Ο άπληστος αλγόριθμος που σχεδιάσαμε δεν αναθέτει ποτέ δύο συγκρουόμενες εργασίες στην ίδια μηχανή.

Θεώρημα: Ο άπληστος αλγόριθμος είναι βέλτιστος.

- Έστω d το πλήθος των διαφορετικών μηχανών στις οποίες αναθέτει εργασία ο αλγόριθμος.

Διαμέριση Διαστημάτων - Ανάλυση Αλγόριθμου

Ο άπληστος αλγόριθμος που σχεδιάσαμε δεν αναθέτει ποτέ δύο συγκρουόμενες εργασίες στην ίδια μηχανή.

Θεώρημα: Ο άπληστος αλγόριθμος είναι βέλτιστος.

- Έστω d το πλήθος των διαφορετικών μηχανών στις οποίες αναθέτει εργασία ο αλγόριθμος.
- Προσθέσαμε τη μηχανή d επειδή μία εργασία, π.χ. η j δεν είναι συμβατή με τις υπόλοιπες $d-1$ μηχανές.

Διαμέριση Διαστημάτων - Ανάλυση Αλγόριθμου

Ο άπληστος αλγόριθμος που σχεδιάσαμε δεν αναθέτει ποτέ δύο συγκρουόμενες εργασίες στην ίδια μηχανή.

Θεώρημα: Ο άπληστος αλγόριθμος είναι βέλτιστος.

- Έστω d το πλήθος των διαφορετικών μηχανών στις οποίες αναθέτει εργασία ο αλγόριθμος.
- Προσθέσαμε τη μηχανή d επειδή μία εργασία, π.χ. η j δεν είναι συμβατή με τις υπόλοιπες $d-1$ μηχανές.
- Αφού τις αναθέτουμε με βάση την χρονική στιγμή έναρξης οι εργασίες με τις οποίες είναι ασύμβατη η j έχουν ώρα έναρξης πριν την s_j.

Διαμέριση Διαστημάτων - Ανάλυση Αλγόριθμου

Ο άπληστος αλγόριθμος που σχεδιάσαμε δεν αναθέτει ποτέ δύο συγκρουόμενες εργασίες στην ίδια μηχανή.

Θεώρημα: Ο άπληστος αλγόριθμος είναι βέλτιστος.

- Έστω d το πλήθος των διαφορετικών μηχανών στις οποίες αναθέτει εργασία ο αλγόριθμος.
- Προσθέσαμε τη μηχανή d επειδή μία εργασία, π.χ. η j δεν είναι συμβατή με τις υπόλοιπες $d - 1$ μηχανές.
- Αφού τις αναθέτουμε με βάση την χρονική στιγμή έναρξης οι εργασίες με τις οποίες είναι ασύμβατη η j έχουν ώρα έναρξης πριν την s_j.
- Άρα έχουμε διενέξεις με d συνολικά εργασίες τη χρονική στιγμή $s_j + \epsilon$.

Διαμέριση Διαστημάτων - Ανάλυση Αλγόριθμου

Ο άπληστος αλγόριθμος που σχεδιάσαμε δεν αναθέτει ποτέ δύο συγκρουόμενες εργασίες στην ίδια μηχανή.

Θεώρημα: Ο άπληστος αλγόριθμος είναι βέλτιστος.

- Έστω d το πλήθος των διαφορετικών μηχανών στις οποίες αναθέτει εργασία ο αλγόριθμος.
- Προσθέσαμε τη μηχανή d επειδή μία εργασία, π.χ. η j δεν είναι συμβατή με τις υπόλοιπες $d-1$ μηχανές.
- Αφού τις αναθέτουμε με βάση την χρονική στιγμή έναρξης οι εργασίες με τις οποίες είναι ασύμβατη η j έχουν ώρα έναρξης πριν την s_j.
- Άρα έχουμε διενέξεις με d συνολικά εργασίες τη χρονική στιγμή $s_j + \epsilon$.

Βασική παρατήρηση: Όλες οι λύσεις χρειάζονται τουλάχιστον d μηχανές.