

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΠΑΡΑΛΛΗΛΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΕ ΠΡΟΒΛΗΜΑΤΑ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

### 5η Εργαστηριακή Άσκηση

Εύρεση θέσης Βασιλισσών σε σκακιέρα που δεν περιέχονται σε κοινή γραμμή στήλη ή διαγώνιο  
(Υλοποίηση γενετικών αλγόριθμων)

## ΠΡΟΕΤΟΙΜΑΣΙΑ

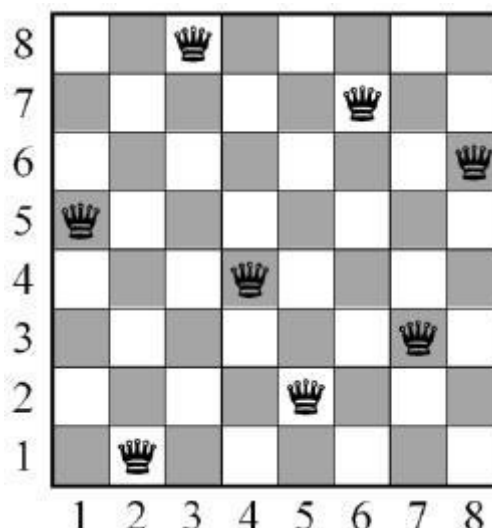
Αρχικά πρέπει να μελετήσετε την βασική δομή των γενετικών αλγορίθμων. Στο e-class υπάρχει αναρτημένη ικανοποιητική βιβλιογραφία και μπορείτε να την βρείτε στην ιστοσελίδα του μαθήματος

<https://eclass.upatras.gr/modules/document/index.php?course=EE975&openDir=/5da1ffa7X42k>

Επιπρόσθετο υλικό μπορείτε να βρείτε στην wikipedia: [https://en.wikipedia.org/wiki/Genetic\\_algorithm](https://en.wikipedia.org/wiki/Genetic_algorithm)

ή εδώ <https://www.geeksforgeeks.org/genetic-algorithms/>

Για την επίλυση του προβλήματος εύρεσης μιας τουλάχιστον θέσης στην οποία οι βασίλισσες έχουν τοποθετηθεί με τέτοιο τρόπο έτσι ώστε καμμία από αυτές να μην συναντά οποιαδήποτε αλλη με οριζόντια, κατακόρυφη ή διαγώνια μετακίνηση, όπως φαίνεται και στο σχήμα που ακολουθεί, υπάρχουν διαθέσιμοι διαφορετικοί αλγόριθμοι



Μπορείτε να ψάξετε στο διαδίκτυο και να βρείτε αυτούς τους αλγόριθμους που βασίζονται σε διαφορετικές τεχνικές όπως και σε γενετικούς αλγόριθμους, πχ.

<https://towardsdatascience.com/genetic-algorithm-vs-backtracking-n-queen-problem-cdf38e15d73f>

Αν θέλετε να μάθετε και κάτι παραπάνω για το πρόβλημα που θα επιλύσετε ή θέλετε να πάρετε μερικές ιδέες για το πως θα υλοποιήσετε τον γενετικό αλγόριθμο ρίξτε μαι ματιά εδώ

<https://www.kaggle.com/mrknoot/genetic-algorithms-solving-the-n-queens-problem>

## ΕΚΦΩΝΗΣΗ

Θα υλοποιήσετε δύο αλγορίθμους επίλυσης του προβλήματος εύρεσης μιας θέσης για την οποία οι  $N$  βασίλισσες τοποθετούνται σε μια σκακιέρα  $N \times N$  τετραγώνων και οι οποίες δεν έχουν κοινή οριζόντια, κατακόρυφη ή διαγώνια γραμμή.

Ο πρώτος και απλούστερος αλγόριθμος πλήρους αναζήτησης ακολουθεί τα εξής βήματα:

Βήμα 1. Τοποθετείται μία βασίλισσα στην θέση (1,1) δηλ. Πρώτη γραμμή, πρώτη στήλη.

Βήμα 2 Στην συνέχεια τοποθετείται η επόμενη βασίλισσα στη επόμενη στήλη και στην σειρά εκείνη που ικανοποιεί τον περιορισμό της μη κοινής γραμμής ή κοινής διαγωνίου με τις βασίλισσες που έχουν ήδη τοποθετηθεί. Αν αυτό είναι εφικτό επαναλαμβάνεται το βήμα 2 μέχρις ότου να τοποθετηθεί η βασίλισσα και στην τελευταία στήλη. Τότε ο αλγόριθμος τερματίζει γιατί έχει βρεθεί μια θέση.

Βήμα 3. Αν δεν μπορεί να τοποθετηθεί η βασίλισσα σε καμμία γραμμή για την τρέχουσα στήλη που μελετάμε τότε γυρνάμε στην προηγούμενη στήλη και μετακινούμε την βασίλισσά της σε μια νέα γραμμή που δεν έρχεται σε σύγκρουση με τις βασίλισσες που είναι τοποθετημένες ήδη. Αν αυτό είναι εφικτό πηγαίνουμε στο βήμα 2, διαφορετικά επαναλαμβάνεται το βήμα 3.

### Εργασία 1

Κατασκευάστε το πρόγραμμα πλήρους αναζήτησης για μια σκακιέρα διαστάσεων  $8 \times 8$ . Υπολογίστε τον αριθμό των βημάτων του αλγόριθμου, δηλαδή πόσες φορές χρειάστηκε να τοποθετήσετε μια βασίλισσα στην σκακιέρα και να ψάξετε αν η θέση της είναι επιτρεπτή, όπως επίσης και τον χρόνο αναζήτησης της λύσης.

### Εργασία 2

Αν και ο αλγόριθμος πλήρους αναζήτησης φαίνεται ισχυρά σειριακός, βρείτε τρόπο να τον κάνετε παράλληλο έτσι ώστε να μειώσετε τον χρόνο αναζήτησης μιας λύσης. Πόσο ταχύτερα βρήκατε την λύση σε σχέση με την υλοποίηση της πρώτης εργασίας; Να είσαστε τίμιοι, δηλαδή να μην χρησιμοποιήσετε την πληροφορία της λύσης/λύσεων της εργασίας 1 για να βρείτε γρήγορα μια λύση στην εργασία 2 ή τις υπόλοιπες.

Αυξήστε το μέγεθος της σκακιέρας στο  $16 \times 16$  ( $N=16$ ) και υπολογίστε τα

βήματα και τους χρόνους στα προγράμματα της εργασίας 1 και 2. Με τα συνήθη υπολογιστικά συστήματα που διαθέτουμε μπορούμε, για αποδεκτούς χρόνους (λιγότερο από 5 λεπτά), να βρούμε λύση για  $N \sim 35$

### Εργασία 3

Υλοποιείτε έναν γενετικό αλγόριθμο σε C ή C++. Αρχικά την σειριακή έκδοση του προγράμματος. Προσαρμόστε τον στο πρόβλημα των N βασιλισσών, δώστε μια συνοπτική και πλήρη περιγραφή του και επιλύστε το πρόβλημα στην σκακιέρα με διαστάσεις 8x8 και 16x16. Δώστε προσοχή στον τρόπο που θα υπολογίσετε την συνάρτηση evaluation, γιατί ο τρόπος με τον οποίο θα ορίσετε πόσο η αυθαίρετη τοποθέτηση των N βασιλισσών πλησιάζει μιά από τις βέλτιστες λύσεις που ψάχνετε θα επηρεάσει σημαντικά την ταχύτητα εύρεσης της μιας βέλτιστης λύσης που αναζητάτε.

### Εργασία 4

Τροποποιείτε το πρόγραμμα της προηγούμενης εργασίας έτσι ώστε να έχετε μια παράλληλη έκδοση του γενετικού αλγόριθμου σε C ή C++ με OpenMP. Δώστε μια συνοπτική και πλήρη περιγραφή του και επιλύστε το πρόβλημα στην σκακιέρα με διαστάσεις 8x8 και 16x16. Συγκρίνετε τους χρόνους υπολογισμών με τις προηγούμενες 3 υλοποιήσεις.

## **ΠΑΡΑΔΟΤΕΑ**

- Για την εργασία 1 στείλτε τον κώδικα σε c ή c++. Να βάλετε σχόλια!!! και τα βήματα και χρόνους που υπολογίσατε
- Για την εργασία 2 στείλτε τον κώδικα σε c ή c++ που υλοποιήσατε, τα βήματα και χρόνους που υπολογίσατε και για τις δύο σκακιέρες.
- Για την εργασία 3, ότι και στην 2. Παρακαλούμε για την σαφή και πλήρη περιγραφή της υλοποίησης του γενετικού αλγόριθμου για να μπορέσουμε να αξιολογήσουμε σωστά τα προγράμματά σας.
- Για την εργασία 4, ότι και στην 3. Παρακαλούμε για την σαφή και πλήρη περιγραφή των αλλαγών που κάνατε έτσι ώστε να παραλληλήσετε τον γενετικό αλγόριθμο. Να συγκρίνετε με τους προηγούμενους χρόνους για να δείτε τι τελικά καταφέρατε.
- Σχολιάστε την προσπάθειά σας. Αν πειραματιστείτε και για μεγαλύτερες σκακιέρες από 16x16, παρακαλώ περιγράψτε και τους χρόνους που πετύχατε.

Καλή χρονιά