

Απόδοση Δυαδικού Δένδρου Έρευνας με Array

1. Μέσος αριθμός συγκρίσεων ανά εισαγωγή:
 $50 = 22,54$
 $10^2 = 25,08$
 $10^3 = 43,79$
 $10^4 = 59,96$
 $10^5 = 88,75$
 $10^6 = 91,56$
2. Μέσος αριθμός συγκρίσεων για 100 τυχαίες αναζητήσεις:
 $50 = 23,12$
 $10^2 = 26,45$
 $10^3 = 42,80$
 $10^4 = 55,34$
 $10^5 = 77,84$
 $10^6 = 82,40$
3. Μέσος αριθμός συγκρίσεων για 100 τυχαίες αναζητήσεις εύρους τιμών:
 $50 : (k=100) = 35,20 , (k=1000) = 69,25$
 $10^2 : (k=100) = 40,40 , (k=1000) = 82,00$
 $10^3 : (k=100) = 70,35 , (k=1000) = 135,95$
 $10^4 : (k=100) = 89,25 , (k=1000) = 179,25$
 $10^5 : (k=100) = 129,65 , (k=1000) = 258,57$
 $10^6 : (k=100) = 137,29 , (k=1000) = 278,47$
4. Συνολικός χρόνος για N εισαγωγές:
 $50 = 1,39 \text{ ms}$
 $10^2 = 2,42 \text{ ms}$
 $10^3 = 17,99 \text{ ms}$
 $10^4 = 104,73 \text{ ms}$
 $10^5 = 1028,18 \text{ ms}$
 $10^6 = 10641,33 \text{ ms}$
5. Συνολικός χρόνος για 100 αναζητήσεις:
 $50 = 0,31 \text{ ms}$
 $10^2 = 0,33 \text{ ms}$
 $10^3 = 0,76 \text{ ms}$
 $10^4 = 0,68 \text{ ms}$
 $10^5 = 0,87 \text{ ms}$
 $10^6 = 0,96 \text{ ms}$

Απόδοση Δυαδικού Δένδρου Έρευνας με Δυναμική Παραχώρηση Μνήμης

6. Μέσος αριθμός συγκρίσεων ανά εισαγωγή:
 $50 = 14,56$
 $10^2 = 16,27$
 $10^3 = 29,53$
 $10^4 = 41,34$
 $10^5 = 61,91$
 $10^6 = 63,44$
7. Μέσος αριθμός συγκρίσεων για 100 τυχαίες αναζητήσεις:
 $50 = 22,61$
 $10^2 = 25,52$
 $10^3 = 42,53$
 $10^4 = 55,73$
 $10^5 = 77,60$
 $10^6 = 83,36$

8. Μέσος αριθμός συγκρίσεων για 100 τυχαίες αναζητήσεις εύρους τιμών:

$$\begin{aligned}50 : (k=100) &= 37,20, (k=1000) = 73,75 \\10^2 : (k=100) &= 39,40, (k=1000) = 78,35 \\10^3 : (k=100) &= 66,90, (k=1000) = 133,70 \\10^4 : (k=100) &= 91,25, (k=1000) = 181,96 \\10^5 : (k=100) &= 133,26, (k=1000) = 259,99 \\10^6 : (k=100) &= 136,37, (k=1000) = 271,54\end{aligned}$$

9. Συνολικός χρόνος για N εισαγωγές:

$$\begin{aligned}50 &= 1,96 \text{ ms} \\10^2 &= 2,67 \text{ ms} \\10^3 &= 19,92 \text{ ms} \\10^4 &= 150,30 \text{ ms} \\10^5 &= 1273,49 \text{ ms} \\10^6 &= 11785,78 \text{ ms}\end{aligned}$$

10. Συνολικός χρόνος για 100 αναζητήσεις:

$$\begin{aligned}50 &= 0,41 \text{ ms} \\10^2 &= 0,33 \text{ ms} \\10^3 &= 0,72 \text{ ms} \\10^4 &= 0,76 \text{ ms} \\10^5 &= 0,89 \text{ ms} \\10^6 &= 0,92 \text{ ms}\end{aligned}$$

Απόδοση δυαδικής αναζήτησης σε ταξινομημένο πίνακα

1. Μέσος αριθμός συγκρίσεων για 100 τυχαίες αναζητήσεις:

$$\begin{aligned}50 &= 31,50 \\10^2 &= 36,40 \\10^3 &= 52,80 \\10^4 &= 70,20 \\10^5 &= 86,20 \\10^6 &= 102,85\end{aligned}$$

2. Μέσος αριθμός συγκρίσεων για 100 τυχαίες αναζητήσεις εύρους τιμών:

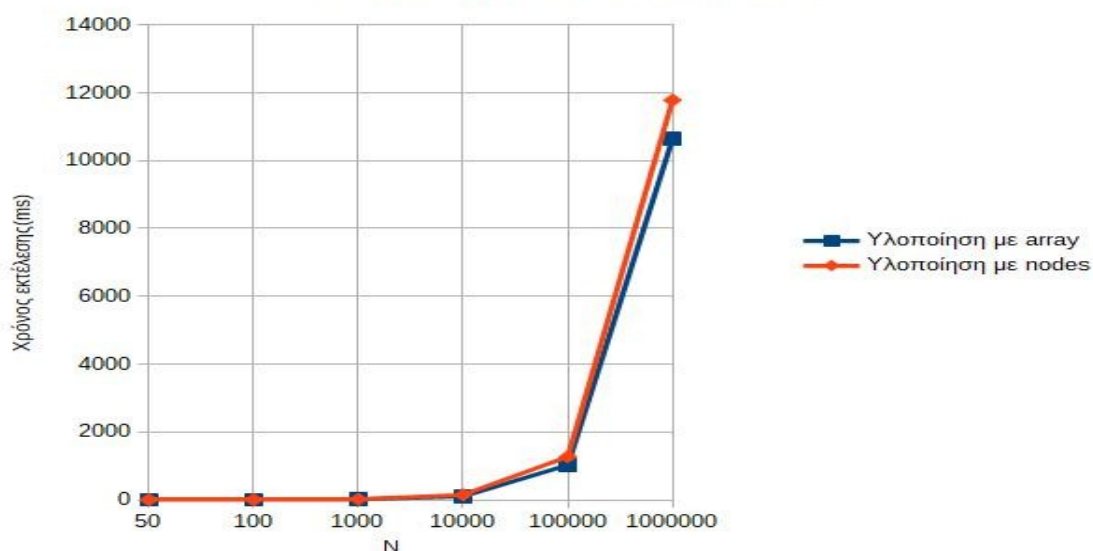
$$\begin{aligned}50 : (k=100) &= 30,61, (k=1000) = 61,39 \\10^2 : (k=100) &= 34,96, (k=1000) = 70,09 \\10^3 : (k=100) &= 48,40, (k=1000) = 96,80 \\10^4 : (k=100) &= 64,20, (k=1000) = 128,36 \\10^5 : (k=100) &= 80,04, (k=1000) = 160,08 \\10^6 : (k=100) &= 93,69, (k=1000) = 188,36\end{aligned}$$

3. Συνολικός χρόνος για 100 αναζητήσεις:

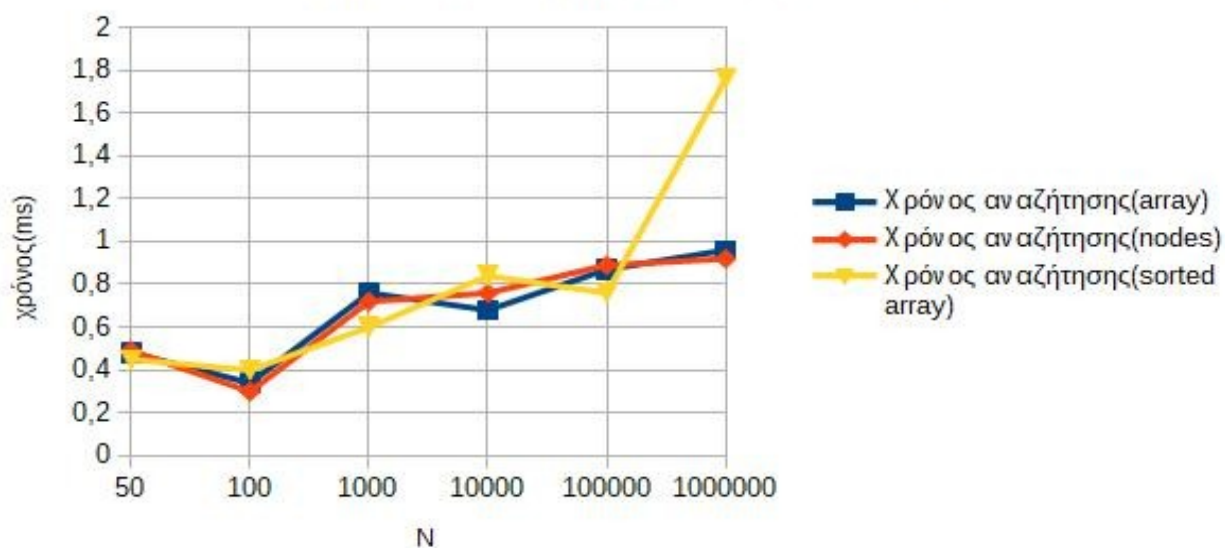
$$\begin{aligned}50 &= 0,45 \text{ ms} \\10^2 &= 0,40 \text{ ms} \\10^3 &= 0,60 \text{ ms} \\10^4 &= 0,84 \text{ ms} \\10^5 &= 0,76 \text{ ms} \\10^6 &= 1,76 \text{ ms}\end{aligned}$$

Μέθοδος	Μέσος αριθμός συγκρίσεων / εισαγωγή	Συνολικός χρόνος για N εισαγωγές	Μέσος αριθμός συγκρίσεων / τυχαία αναζήτηση	Συνολικός χρόνος για 100 αναζητήσεις	Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=100)	Μέσος αριθμός συγκρίσεων / αναζήτηση εύρους (K=1.000)
ΔΔΕ με δυναμική παραχώρηση μνήμης	63,44	11785,78 ms	83,36	0,92 ms	136,37	271,54
ΔΔΕ με array	91,56	10641,33 ms	82,40	0,96 ms	137,29	278,47
Ταξινομημένο πεδίο	■	■	102,85	1,76 ms	93,67	188,36

Συνολικός χρόνος για N εισαγωγές



Συνολικός χρόνος αναζήτησης 100 τυχαίων στοιχείων



Για την εισαγωγή N στοιχείων:

- Η υλοποίηση με array για $N=50$ είχε μέσο όρο 22,54 προσβάσεις και συνολικό χρόνο $t=1,39$ ms ενώ για $N=10^6$ είχε 91,56 προσβάσεις και χρόνο $t=10641$ ms.
- Η υλοποίηση με δυναμική παραχώρηση μνήμης για $N=50$ είχε μέσο όρο 14,56 προσβάσεις και συνολικό χρόνο $t=1,96$ ms και για $N=10^6$ είχε 63,44 προσβάσεις και χρόνο $t=11785$ ms.

Άρα για αύξηση του N κατά 20000($50 \rightarrow 10^6$) είχαμε μια αύξηση του t κατά 7655 για την πρώτη και 6012 για την δεύτερη, δηλαδή περίπου γραμμική αύξηση. Επομένως συμπεραίνουμε ότι στην εισαγωγή N στοιχείων παρατηρούμε μια γραμμική αύξηση στον συνολικό χρόνο εισαγωγής και για τις δύο υλοποιήσεις, με την δυναμική υλοποίηση να είναι πιο αργή αλλά να έχει σαφώς λιγότερο μέσο όρο προσβάσεων. Η στατική δομή των πινάκων επιτρέπει την γρηγορότερη εισαγωγή νέων αριθμών σε αυτούς αλλά το πληρώνουμε αυτό σε αριθμό προσβάσεων σε σύγκριση με την δυναμική μνήμη.

Για την αναζήτηση 100 τυχαίων στοιχείων:

- Η υλοποίηση με array για $N=50$ είχε κατά μέσο όρο 23,12 προσβάσεις και χρόνο $t=0,31$ ms ενώ για $N=10^6$ είχε 82,4 προσβάσεις και χρόνο $t=0,96$ ms.
- Για δυναμική παραχώρηση μνήμης για $N=50$ είχε κατά μέσο όρο 22,61 προσβάσεις και $t=0,41$ ms και για $N=10^6$ είχε 83,36 προσβάσεις και χρόνο $t=0,92$ ms.
- Για υλοποίηση με ταξινομημένο πίνακα για $N=50$ είχε μέσο όρο 31,5 προσβάσεις και $t=0,45$ ms και για $N=10^6$ είχε 102,85 προσβάσεις μέσο όρο και $t=1,76$ ms.

Αρχικά παρατηρούμε και για τις τρεις μια λογαριθμική αύξηση των χρόνων αναζήτησης από το μικρότερο στο μεγαλύτερο N ($\log 20000 \sim 4,3$) με τις υλοποιήσεις με στατική και δυναμική παραχώρηση μνήμης να έχουν παρόμοιους χρόνους ενώ η υλοποίηση με ταξινομημένο πίνακα ήταν σαφώς πιο αργή.

Συγκεκριμένα παρατηρούμε ότι η υλοποίηση με ταξινομημένο πίνακα είναι σαφώς πιο αργή και με περισσότερες προσβάσεις από τις άλλες δύο. Αυτό την καθιστά εντελώς μη αποδοτική. Όσον αφορά τις υπόλοιπες δύο υλοποιήσεις έχουν παρόμοιο αριθμό προσβάσεων όσο και ταχύτητα εκτέλεσης άρα είναι περίπου το ίδιο αποδοτικές για αναζήτηση. Τα αποτελέσματα για τους συνολικούς χρόνους για αναζήτηση 100 τυχαίων στοιχείων δεν είχαν καλή αξιοπιστία αλλά δίνουν την γενική εικόνα.

Για την αναζήτηση εύρους(range search):

Συγκρίνοντας τους αριθμούς προσβάσεων των τριών υλοποιήσεων παρατηρούμε ότι η υλοποίηση με ταξινομημένο πίνακα είναι η πιο αποδοτική ενώ οι άλλες δύο λιγότερο αποδοτικές. Οι υλοποιήσεις με στατική και δυναμική παραχώρηση μνήμης έχουν περίπου τον ίδιο αριθμό προσβάσεων. Η καλύτερη αποδοτικότητα του ταξινομημένου πίνακα πιστεύω ότι οφείλεται στο χαρακτηριστικό του, της σειριακής ταξινόμησης η οποία υπερτερεί χωρικά στην αναζήτηση αριθμών σε ένα συγκεκριμένο εύρος σε αντίθεση με την αναζήτηση μεμονωμένου αριθμού.