

Μάριος Κωνσταντίνος Κωνσταντάκης, csd3219@csd.uoc.gr  
A.M: 3219.

Το πρόγραμμα μου αρχικά παίρνει το αρχείο με τα σώματα. Δημιουργώ μία δομή με όλα τα σώματα και εκτελώ τον αναδρομικό αλγόριθμο που δημιουργεί ένα τετραδικό διπλά συνδεδεμένο δέντρο που συμβολίζει το σύμπαν.

Σε κάθε αναδρομική κλήση, ο αλγόριθμος μου διαιρεί τον χώρο σε 4 τμήματα, και αντίστοιχα δημιουργεί 4 κόμβους που αντιστοιχούν σε κάθε ένα τμήμα. Παράλληλα, υπολογίζω ποια σώματα αντιστοιχίζονται σε κάθε τμήμα και καλώ αναδρομικά την συνάρτηση δίνοντας το αντίστοιχο σύνολο σωμάτων ως όρισμα για να το διαιρέσει και αυτό σε 4 τμήματα και ούτω κάθε εξής.

Ο αλγόριθμος τερματίζει την αναδρομική του κλήση όταν καταλήξουμε σε υπο-σύμπαν το οποίο περιέχει 0 ή 1 σώμα. Κατά την διάρκεια της δημιουργίας του δέντρου υπολογίζω το κέντρο μάζας και τα X, Y του εσωτερικών κόμβων.

Μετά την δημιουργία του αρχικού δέντρου, ο αλγόριθμος εκτελεί μία διάσχιση στο δέντρο και για κάθε σώμα υπολογίζει την συνισταμένη δύναμη(netForce) που δένεται από τα άλλα σώματα ή σύμπαντα και έπειτα υπολογίζει την επιτάχυνση και την δύναμη που δέχεται σε κάθε άξονα. Η δύναμη αυτή κινεί το σώμα και υπολογίζω την νέα θέση του X, Y στο χώρο. Έπειτα, η όλη διαδικασία επαναλαμβάνεται και δημιουργώ ένα νέο δέντρο που περιλαμβάνει τους κόμβους στις νέες τους θέσεις. Αυτό επαναλαμβάνεται repetition φορές, ανάλογα με την είσοδο του χρήστη. Στο τέλος αποτυπώνω το τελικό δέντρο που έχει προκύψει σε αρχείο.

Οι επαναλήψεις(repetitions) είναι 100 και 1000, και δίνονται από τον χρήστη, ως όρισμα από την γραμμή εντολών.

Ως αρχεία εισόδου, χρησιμοποιήθηκαν τα 5 περσινά input1-5.txt

Ο αριθμός των threads δίνεται ως όρισμα από την γραμμή εντολών, και όπως ζητήθηκε από την εκφώνηση, ισούται με 1, 2 ή 4.

Έβγαλα τα κειμενάκια που σχολίαζαν κάτω κάτω σε κάθε αρχείο εισόδου, επειδή υπήρχε θέμα με την scanner.next() στην Java.

Το πρόγραμμα τυπώνει όλη την ενδιάμεση διαδικασία, αλλά δημιουργεί κιόλας το τελικό αρχείο output.txt όπως ζητήθηκε, που περιέχει τις τελικές θέσεις, ταχύτητες των σωμάτων κλπ.

Μετρώ την ώρα εκτέλεσης με την εντολή time του command line. Έχω συμπεριλάβει κώδικα στις υλοποιήσεις μου, για την μέτρηση του χρόνου εκτέλεσης, αλλά δεν νομίζω ότι είναι τόσο ακριβές όσο η time.

Η εντολή είναι:

```
<executable> repetitions threadCount input.txt output.txt
```

δλδ,

```
time Main repetitions threadCount input.txt output.txt (C++)
```

&

```
time java Main repetitions threadCount input.txt output.txt (Java)
```

```
πχ time Main 100 1 ../../input1.txt output.txt
```

```
time java Main 100 1 ../../input1.txt output.txt
```

Το μηχάνημα μου έχει AMD Ryzen 5 επεξεργαστή & 16GB RAM.

Αναλυτικότερα:

CPU

AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx

Ταχύτητα βάσης: 2,10 GHz  
Υποδοχές: 1  
Πυρήνες: 4  
Λογικοί επεξεργαστές: 8  
Λειτουργία αναπαράστασης: Ενεργοποιημένο  
Cache L1: 384 KB  
Cache L2: 2,0 MB  
Cache L3: 4,0 MB

Χρήση 2%  
Ταχύτητα 1,87 GHz  
Χρόνος λειτουργίας 15:18:23:04  
Διεργασίες 193  
Νήματα 2287  
Δείκτες χειρισμού 90198

Ακολουθούν τα screenshots για κάθε εκτέλεση.

Σειριακά προγράμματα:

C++:

input1.txt

```
#####  
The program has been running for 0.021805 seconds, with repetitions=100, testfile=../input1.txt.  
  
real    0m0.127s  
user    0m0.005s  
sys     0m0.020s  
csd3219@kos:~/ask34/ask3$ time Main 100 ../input1.txt output.txt
```

```
#####  
The program has been running for 0.168086 seconds, with repetitions=1000, testfile=../input1.txt.  
  
real    0m1.023s  
user    0m0.086s  
sys     0m0.087s  
csd3219@kos:~/ask34/ask3$ time Main 1000 ../input1.txt output.txt
```

```
#####
The program has been running for 1.840578 seconds, with repetitions=10000, testfile=./input1.txt.

real    0m11.008s
user    0m0.907s
sys     0m0.941s
csd3219@kos:~/ask34/ask3$ time Main 10000 ../input1.txt output.txt
```

input2.txt

```
#####
The program has been running for 0.477872 seconds, with repetitions=100, testfile=./input2.txt.

real    0m2.221s
user    0m0.249s
sys     0m0.234s
csd3219@kos:~/ask34/ask3$ time Main 100 ../input2.txt output.txt
```

```
#####
The program has been running for 4.517085 seconds, with repetitions=1000, testfile=./input2.txt.

real    0m27.906s
user    0m2.234s
sys     0m2.294s
csd3219@kos:~/ask34/ask3$ time Main 1000 ../input2.txt output.txt
```

input3.txt

```
#####
The program has been running for 0.374605 seconds, with repetitions=100, testfile=./input3.txt.

real    0m0.444s
user    0m0.184s
sys     0m0.195s
csd3219@kos:~/ask34/ask3$ time Main 100 ../input3.txt output.txt
```

```
#####
The program has been running for 3.257954 seconds, with repetitions=1000, testfile=./input3.txt.

real    0m20.145s
user    0m1.672s
sys     0m1.596s
csd3219@kos:~/ask34/ask3$ time Main 1000 ../input3.txt output.txt
```

input4.txt

```
#####
The program has been running for 3.992869 seconds, with repetitions=100, testfile=./input4.txt.

real    0m24.094s
user    0m1.958s
sys     0m2.050s
csd3219@kos:~/ask34/ask3$ time Main 100 ../input4.txt output.txt
```

```
#####
The program has been running for 37.401306 seconds, with repetitions=1000, testfile=./input4.txt.

real    3m41.833s
user    0m18.486s
sys     0m18.972s
csd3219@kos:~/ask34/ask3$ time Main 1000 ../input4.txt output.txt
```

input5.txt

```
#####
The program has been running for 3.154337 seconds, with repetitions=100, testfile=./input5.txt.

real    0m18.935s
user    0m1.523s
sys     0m1.651s
csd3219@kos:~/ask34/ask3$ time Main 100 ../input5.txt output.txt
```

```
#####
The program has been running for 29.673882 seconds, with repetitions=1000, testfile=./input5.txt.

real    3m32.498s
user    0m15.140s
sys     0m14.575s
csd3219@kos:~/ask34/ask3$ time Main 1000 ../input5.txt output.txt
```

Java:

input1.txt

```
Elapsed Time: 300857387 nanoseconds
```

```
real    0m0.455s
user    0m0.801s
sys     0m0.101s
csd3219@kos:~/ask34/ask4$ time java Main 100 ../input1.txt output.txt
```

```
Elapsed Time: 690228249 nanoseconds
```

```
real    0m0.784s
user    0m1.512s
sys     0m0.601s
csd3219@kos:~/ask34/ask4$ time java Main 1000 ../input1.txt output.txt
```

```
Elapsed Time: 4567896041 nanoseconds
```

```
real    0m4.654s
user    0m3.630s
sys     0m1.974s
csd3219@kos:~/ask34/ask4$ time java Main 10000 ../input1.txt output.txt
```

input2.txt

```
Elapsed Time: 1615166028 nanoseconds
```

```
real    0m1.709s
user    0m2.584s
sys     0m0.946s
```

```
csd3219@kos:~/ask34/ask4$ time java Main 100 ../input2.txt output.txt █
```

```
Elapsed Time: 15796149036 nanoseconds
```

```
real    0m15.877s
user    0m7.428s
sys     0m7.511s
```

```
csd3219@kos:~/ask34/ask4$ time java Main 1000 ../input2.txt output.txt █
```

```
Elapsed Time: 246564965658 nanoseconds
```

```
real    4m6.665s
user    0m51.250s
sys     1m12.577s
```

```
csd3219@kos:~/ask34/ask4$ time java Main 10000 ../input2.txt output.txt █
```

input3.txt

```
Elapsed Time: 1085654125 nanoseconds
```

```
real    0m1.191s
user    0m1.854s
sys     0m0.578s
```

```
csd3219@kos:~/ask34/ask4$ time java Main 100 ../input3.txt output.txt █
```

```
Elapsed Time: 11549996105 nanoseconds
```

```
real    0m11.644s
user    0m4.727s
sys     0m4.225s
```

```
csd3219@kos:~/ask34/ask4$ time java Main 1000 ../input3.txt output.txt █
```

```
Elapsed Time: 180162057501 nanoseconds
```

```
real    3m0.262s
user    0m33.270s
sys     0m44.530s
```

```
csd3219@kos:~/ask34/ask4$ time java Main 10000 ../input3.txt output.txt █
```

input4.txt

```
Elapsed Time: 15858726741 nanoseconds

real    0m15.967s
user    0m7.862s
sys     0m8.107s
csd3219@kos:~/ask34/ask4$ time java Main 100 ../input4.txt output.txt
```

```
Elapsed Time: 158389219636 nanoseconds

real    2m38.503s
user    0m58.123s
sys     1m19.050s
csd3219@kos:~/ask34/ask4$ time java Main 1000 ../input4.txt output.txt
```

input5.txt

```
Elapsed Time: 20625302762 nanoseconds

real    0m20.734s
user    0m6.472s
sys     0m15.431s
csd3219@kos:~/ask34/ask4$ time java Main 100 ../input5.txt output.txt
```

```
Elapsed Time: 111066942345 nanoseconds

real    1m51.174s
user    0m43.209s
sys     0m57.325s
csd3219@kos:~/ask34/ask4$ time java Main 1000 ../input5.txt output.txt
```

Παράλληλα προγράμματα:

Java:

input1.txt

```
real    0m0.480s
user    0m0.829s
sys     0m0.102s
csd3219@anafiki:~/ask34/ask4/parallel$ time java Main 100 1 ../../input1.txt output.txt
```

```
real    0m0.431s
user    0m0.958s
sys     0m0.258s
csd3219@anafiki:~/ask34/ask4/parallel$ time java Main 100 2 ../../input1.txt output.txt
```

```
real    0m0.645s
user    0m1.315s
sys     0m0.379s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 100 4 ../../input1.txt output.txt
```

```
real    0m1.112s
user    0m2.039s
sys     0m0.301s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 1000 1 ../../input1.txt output.txt
```

```
real    0m0.688s
user    0m1.451s
sys     0m0.422s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 1000 2 ../../input1.txt output.txt
```

```
real    0m1.187s
user    0m1.667s
sys     0m0.338s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 1000 4 ../../input1.txt output.txt
```

input2.txt

```
real    0m2.490s
user    0m2.435s
sys     0m0.953s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 100 1 ../../input2.txt output.txt
```

```
real    0m0.353s
user    0m0.690s
sys     0m0.131s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 100 2 ../../input2.txt output.txt
```

```
real    0m9.332s
user    0m5.979s
sys     0m7.212s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 100 4 ../../input2.txt output.txt
```

```
real    0m27.944s
user    0m6.967s
sys     0m14.790s
csd3219@anafifi:~/ask34/ask4/parallel$ time java Main 1000 1 ../../input2.txt output.txt
```

```
real    0m0.357s
user    0m0.754s
sys     0m0.139s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 1000 2 ../../input2.txt output.txt
```

```
real    0m0.576s
user    0m1.298s
sys     0m0.260s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 1000 4 ../../input2.txt output.txt
```

input3.txt

```
real    0m1.364s
user    0m1.916s
sys     0m0.894s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 100 1 ../../input3.txt output.txt
```

```
real    0m0.315s
user    0m0.602s
sys     0m0.122s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 100 2 ../../input3.txt output.txt
```

```
real    0m8.075s
user    0m5.109s
sys     0m4.704s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 100 4 ../../input3.txt output.txt
```

```
real    0m15.183s
user    0m4.369s
sys     0m4.370s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 1000 1 ../../input3.txt output.txt
```

```
real    0m0.374s
user    0m0.692s
sys     0m0.109s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 1000 2 ../../input3.txt output.txt
```

```
real    0m4.574s
user    0m3.528s
sys     0m3.685s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 1000 4 ../../input3.txt output.txt
```

input4.txt

```
real    0m18.688s
user    0m7.290s
sys     0m7.349s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 100 1 ../../input4.txt output.txt
```



```
real    0m0.782s
user    0m1.390s
sys     0m0.518s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 100 2 ../../input4.txt output.txt
```

```
real    0m0.500s
user    0m1.379s
sys     0m0.139s
csd3219@anafi:~/ask34/ask4/parallel$ time java Main 100 4 ../../input4.txt output.txt
```

```
real    3m30.879s
user    0m53.122s
sys     1m43.109s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 1000 1 ../../input4.txt output.txt
```

```
real    0m0.732s
user    0m1.479s
sys     0m0.449s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 1000 2 ../../input4.txt output.txt
```

```
real    0m1.686s
user    0m1.609s
sys     0m0.672s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 1000 4 ../../input4.txt output.txt
```

input5.txt

```
real    0m16.398s
user    0m6.206s
sys     0m6.691s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 100 1 ../../input5.txt output.txt
```

```
real    0m13.434s
user    0m6.730s
sys     0m6.489s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 100 2 ../../input5.txt output.txt
```

```
real    0m6.913s
user    0m4.062s
sys     0m4.046s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 100 4 ../../input5.txt output.txt
```

```
real    2m43.653s
user    0m40.278s
sys     1m22.969s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 1000 1 ../../input5.txt output.txt
```

```
real    0m0.478s
user    0m1.145s
sys     0m0.212s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 1000 2 ../../input5.txt output.txt
```

```
real    0m0.891s
user    0m1.586s
sys     0m0.574s
csd3219@limnos:~/ask34/ask4/parallel$ time java Main 1000 4 ../../input5.txt output.txt
```

Ερωτήσεις θεωρίας:

1) Ξέρω ότι  $\text{speedup} = \text{χρόνος εκτέλεσης σειριακό πρόγραμμα} / \text{χρόνος εκτέλεσης παράλληλο πρόγραμμα}$ .

Θεωρώ πως το average speedup είναι το άθροισμα των διαφορών της παράλληλης εκδοχής από την σειριακή εκδοχή.

ή

$\text{average speedup} = \Sigma (\text{παράλληλη εκδοχή} - \text{σειριακή εκδοχή}) / \text{αριθμός μετρήσεων}$ .

$\text{speedup}(\text{rep}=100, \text{input1.txt}, 1 \text{ thread}) = 0,455/0,480 = 0,94$ .  
 $\text{speedup}(\text{rep}=100, \text{input1.txt}, 2 \text{ threads}) = 0,455/0,431 = 1,05$ .  
 $\text{speedup}(\text{rep}=100, \text{input1.txt}, 4 \text{ threads}) = 0,455/0,645 = 0,70$ .

$\text{speedup}(\text{rep}=1000, \text{input1.txt}, 1 \text{ thread}) = 0,455/1,112 = 0,40$ .  
 $\text{speedup}(\text{rep}=1000, \text{input1.txt}, 2 \text{ threads}) = 0,455/0,668 = 0,68$ .  
 $\text{speedup}(\text{rep}=1000, \text{input1.txt}, 4 \text{ threads}) = 0,455/1,187 = 0,38$ .

$\text{speedup}(\text{rep}=100, \text{input2.txt}, 1 \text{ thread}) = 1,709/2,490 = 0,68$ .  
 $\text{speedup}(\text{rep}=100, \text{input2.txt}, 2 \text{ threads}) = 1,709/0,353 = 4,81$ .  
 $\text{speedup}(\text{rep}=100, \text{input2.txt}, 4 \text{ threads}) = 1,709/9,332 = 0,18$ .

$\text{speedup}(\text{rep}=1000, \text{input2.txt}, 1 \text{ thread}) = 1,709/27,944 = 0,06$ .  
 $\text{speedup}(\text{rep}=1000, \text{input2.txt}, 2 \text{ threads}) = 1,709/0,357 = 4,78$ .  
 $\text{speedup}(\text{rep}=1000, \text{input2.txt}, 4 \text{ threads}) = 1,709/0,576 = 2,96$ .

Και ούτω καθεξής.

Το average speedup μπορεί να είναι και το άθροισμα τους διά το πλήθος όλων των επιμέρους speedups.

Πχ για input1-2.txt

$\text{avg} = (0,94 + 1,05 + 0,70 + 0,40 + 0,68 + 0,38 + 0,68 + 4,81 + 0,18 + 0,06 + 4,78 + 2,96) / 12 = \dots$

$\text{average speedup} = (0,94 + 1,05 + 0,70 + 0,40 + 0,68 + 0,38 + 0,68 + 4,81 + 0,18 + 0,06 + 4,78 + 2,96 + \dots) / 6$   
μετρήσεις για 5 αρχεία  $= (0,94 + 1,05 + 0,70 + 0,40 + 0,68 + 0,38 + 0,68 + 4,81 + 0,18 + 0,06 + 4,78 + 2,96 + \dots) / 30 = \dots$

2)

3) Βλέπουμε ότι οι 2 Java εκδοχές(σειριακή & παράλληλη) είναι πολύ γρηγορότερες των αντίστοιχων C++ υλοποιήσεων.