ΔΕΥΤΕΡΗ ΕΡΓΑΣΙΑ ΣΤΙΣ ΔΟΜΕΣ ΔΕΔΟΜΕΝΩΝ

ΜΠΙΛΙΑΣ ΓΕΩΡΓΙΟΣ 3200278

ΜΠΟΥΡΑΖΑΝΑΣ ΠΑΝΑΓΙΩΤΗΣ 3200268

Αρχικά, για την Greedy ουσιαστικά χρησιμοποιήσαμε μια μέθοδο **ReadFile** οπού παίρνει ως είσοδο ένα αρχείο **txt** για να το διαβάσει. Το διάβασμα του αρχείου το κάναμε ανά γραμμή με την χρήση της μεθόδου **ReadLine()**. Έτσι, επειδή γνωρίζουμε πως οι 2 πρώτες σειρές θα αποτελούνται από ένα νούμερο η κάθε μια(η πρώτη είναι ο αριθμός των **processors** και η δεύτερη ο αριθμός των **tasks**), απλά διαβάζουμε κάθε γραμμή και τις αποθηκεύουμε σε μια μεταβλητή int την κάθε μια ξεχωριστά με την χρήση της μεθόδου **Integer.parseInt(String)**. Η πρώτη μεταβλητή (**pronum**) όπως προαναφέραμε θα είναι ο αριθμός των επεξεργαστών ενώ η δεύτερη (**tasknum**) ο αριθμός των διεργασιών(**tasknum**). Στην συνέχεια, χρησιμοποιήσαμε μια **while loop** οπού όσο η γραμμή είναι διαφορετική του κενού **(!= null**) μπαίνει μέσα στην loop και αν ο αριθμός των διεργασιών(**tasks**) εξακολουθεί να είναι μικρότερος από των γραμμών διεργασιών(**linenum**), η **linenum** είναι μια σαν counter που θέσαμε να αυξάνεται ανά 1 όσο προχωράει γραμμή ο **Reader** μέσα στην **loop** για να παίρνουμε τον αριθμό των διεργασιών που έχουν γραφτεί και να τον συγκρίνουμε με αυτόν που δηλώθηκε στην $2^{η}$ γραμμή, που έχουν διαβαστεί, αφαιρούμε το κενό αναμεσά στα 2 νούμερα και κρατάμε με την χρήση **split** και σε ένα πίνακα **word** τους δυο αριθμούς που υπάρχουν σε **string**. Ο πρώτος είναι το **id** του **Task** και ο δεύτερος ο χρόνος επεξεργασίας (**time**) του. Στην συνέχεια θέτουμε σε int μεταβλητές τα 2 στοιχεία του πίνακα με ανάλογο τρόπο όπως και τις προηγούμενες(**Integer.parseInt)**, δηλαδή θέτουμε το **word[0]** ως **id** και το **word[1]** ως **time**. Έτσι, μετα δημιουργούμε το αντικείμενο **Task** οπού βάζουμε το **id** και τον χρόνο επεξεργασίας(**time)** μέσα του και το αποθηκεύουμε σε έναν πίνακα που θα έχει μέσα του όλα αυτά τα αντικείμενα τύπου **Task** που δημιουργούνται μέσα σ' αυτό το **loop**. Μόλις τελειώσει η **loop** κλείνει και ο **Reader** (**close()**).

Έτσι, στην συνέχεια, αν ο αριθμός των διεργασιών(**tasknum**) είναι διαφορετικός απ' αυτόν των γραμμών διεργασιών(**linenum**) τότε εκτυπώνει ανάλογο μήνυμα και δεν συνεχίζεται η διαδικασία (κλείνει το πρόγραμμα), αλλιώς συνεχίζει και όσο το **i** που κάνει **loop** μέχρι να είναι ίσο με το αριθμό των διεργασιών(**tasknum**) είναι μικρότερο απ' αυτόν των επεξεργαστών δημιουργεί μια λίστα (λίστα του επεξεργαστή) οπού παίρνει το ανάλογο στοιχείο του πίνακα(αρχίζει απτό πρώτο και κάθε φορά πηγαίνει στο επόμενο στοιχείο), το βάζει εκεί μέσα και δημιουργεί ένα αντικείμενο **Processor** οπού μέσα του έχει αυτή την λίστα που μόλις δημιουργήσαμε γι' αυτόν καθώς και ένα **id** που του έχουμε θέσει ως το i ώστε να ξεκιναει απ' το 1 και να αυξανεται κατά 1. Στην συνέχεια, βάζουμε αυτόν τον Processor στην ουρά προτεραιότητας. Όταν το **i** γίνει μεγαλύτερο του αριθμού των επεξεργαστών (**pronum**) αρχίζει η διαδικασία της ουράς προτεραιότητας(**MaxPQ**).

Ουσιαστικά αυτό που κάνουμε με την ουρά προτεραιότητας είναι ότι παίρνουμε τον Επεξεργαστή με την μέγιστη προτεραιότητα (δηλαδή αυτόν που έχει το μικρότερο συνολικό **time**) με την χρήση της **getMax()**, παίρνουμε την λίστα του(**getList()**) και βάζουμε μέσα σε αυτή την ανάλογη διεργασία απ' τον πίνακα. Στην συνέχεια, επιστρέφουμε τον Processor στην ουρά προτεραιότητας με την χρήση της **insert(Processor x)** και παίρνει την ανάλογη θέση που του αξίζει μετα την ένταξη της διεργασίας(**ptask**) στην λίστα του.

Στην συνέχεια, όσο η μέγιστη προτεραιότητα απτή ουρά προτεραιότητας δεν είναι κενή, παίρνει τον επεξεργαστή με την μέγιστη προτεραιότητα, τον κρατάει σε μια μεταβλητή

τύπου **Processor** και τον αφαιρεί με την χρήση της **getmax()**, παίρνει την **makespan** του με την χρήση της **getActiveTime()** και την συγκρίνει με το **max** που έχουμε αρχικοποιήσει ως 0 και ανάλογα αλλάζει. Έτσι, βρίσκουμε το μέγιστο **makespan**. Αν ο αριθμός των διεργασιών (**tasknum**) είναι μικρότερος του 50 μέσα στο loop τυπώνει το **id** του κάθε επεξεργαστή μαζί με το **makespan** του και την λίστα με τις διεργασίες του(με χρήση της αντίστοιχων **get()**). Και στην συνέχεια μόλις βγει απτό **loop** απλά εκτυπώνει το μέγιστο **makespan**.

Όσο αφορά το μέρος Γ της εργασίας ο αλγόριθμος ταξινόμησης που ακολουθήσαμε είναι η Heapsort. Επειδή αποθηκεύσαμε αρχικά όλα τα **Tasks** σε πίνακα ακολουθήσαμε την ταξινόμηση σε πίνακα για να μην δημιουργήσουμε εξτρά πολυπλοκότητα βάζοντας τα στοιχεία σε λίστα καθώς και επειδή η υλοποίηση ταξινόμησης σε πίνακα μας φάνηκε σχετικά πιο γνώριμη και λόγω πιο εύκολης ανακατεύθυνσης σε οποιοδήποτε μέρος του πίνακα την θεωρήσαμε γενικά πιο προσιτή.

Με ένα τρέξιμο 30 τυχαίων αρχείων **txt** που δημιουργούνται με το τρέξιμο της Comparisons, οπού, πιο συγκεκριμένα, δημιουργούνται **10 αρχεία ανά N** (δηλαδή για **3 N**) και **N** είναι ο αριθμός των διεργασιών που θα έχει το κάθε **file**, κατευθείαν έχουμε ένα εύλογο συμπέρασμα. Το συμπέρασμα είναι ότι και στις 3 περιπτώσεις ο μέσος ορός των **makespan** του κάθε αρχείου μετα την ταξινόμηση είναι συγκριτικά μικρότερος απ' αυτόν χωρίς ταξινόμηση. Δηλαδή, ο **αλγόριθμος 2** κάνει συγκριτικά μικρότερο το **average makespan** απ' αυτό με την χρήση του πρώτου στα ιδιά **files** και συνεπώς είναι πιο αποδοτικός. Πιο συγκεκριμένα, με βάση το παράδειγμα έχουμε:

Για N = 100:

Average makespan:	1071
Average sorted makespan:	986

(8.26% διαφορά)

Για N = 250:

Average makespan:	1774
Average sorted makespan:	1680

(5.44% διαφορά)

Για N = 500:

Average makespan:	2331
Average sorted makespan:	2237

(4.11% διαφορά)

Αναλυτικότερα, παρατηρούμε μια διαφορά της τάξης από 3 έως 10% που μπορεί να μην φαίνεται ιδιαίτερα μεγάλη αλλά στην υλοποίηση μεγάλων προγραμμάτων μια τέτοια διαφορά θα ήταν αρκετά αισθητή.

ΥΓ: τα αποτελέσματα αυτά είναι ενδεικτικά και αναμένεται να έχετε διαφορετικά όταν τρέξετε τον κώδικα. Αυτό οφείλεται στο γεγονός πως κάθε

φορά που τρέχει η Comparisons δημιουργεί 30 καινούργια Files για τα οποία κάνει την μέθοδο ή αντικαθιστά τα προηγούμενα εάν δηλώσουμε το ίδιο όνομα.

Για το τρέξιμο των αρχείων από IDE:

Για την **Greedy** το μόνο που έχετε να κάνετε είναι να πάτε στο **Run > Run Configurations… > Arguments** και μέσα στο **Program arguments** να γράψετε το **path** του αρχείου που θέλετε να διαβάσετε. Πχ. Εγώ αν θέλω να διαβάσω το αρχείο **merosb.txt** θα πάω εκεί που προανέφερα και θα γράψω **C:\Users\georg\eclipse-workspace\3200278_3200268\src\Data\merosb.txt**

Για την **Comparisons** θα πάτε ακριβώς στο ίδιο μέρος όπως και στην **Greedy** απλά θα γράψετε το **path** που θέλετε να αποθηκευτούν τα αρχεία **+ το όνομα απτό οποίο θέλετε να ξεκινάνε τα αρχεία**. Το πρόγραμμα έχει φτιαχτεί έτσι ώστε από μόνο του να δίνει στο όνομα του αρχείου το ανάλογο νούμερο. Δηλαδή, εγώ για παράδειγμα αν πάω εκεί και γράψω "**C:\Users\georg\eclipse-workspace\3200278_3200268\src\Data\File**", το πρόγραμμα θα πάει μέσα στον φάκελο Data του path και θα δημιουργήσει ξεχωριστά αρχεία με ονόματα **File1.txt, File2.txt, File3.txt….,File30.txt** και θα τα τρέξει κατευθείαν.

Για το cmd:

Αρχικά θα κάνετε **cd** και θα βάλετε ως **path** το φάκελο src του project. Πχ. Εγώ βάζω **C:\Users\georg\eclipse-workspace\3200278_3200268\src**. Στην συνέχεια θα κάνετε compile όλα τα αρχεία java κάνοντας **javac *.java**.

Μετα για την **Greedy** αυτό που θα κάνετε είναι να γράψετε **java Greedy** και το **path** του αρχείου που θέλετε να διαβάσετε. Δηλαδή, εγώ για παράδειγμα για να διαβάσω το αρχείο **merosb.txt** που βρίσκεται στον φάκελο **Data** θα κάνω:

java Greedy C:\Users\georg\eclipse-workspace\3200278_3200268\src\Data\merosb.txt

*Σε περίπτωση που θέλετε να τεστάρετε με ταξινομημένο πίνακα απτή **Greedy** απλά αλλάξτε το **String choice = "no sort"** σε **sort**. Μπορείτε και με οποιοδήποτε άλλο τρόπο να γράψετε το sort εκτός από με κενά.(λόγω της **IgnoreCase**)*

Για την **Comparisons** θα λειτουργήσω με ανάλογο τρόπο. Δηλαδή, θα κάνω:

java Comparisons C:\Users\georg\eclipse-workspace\3200278_3200268\src\Data\File και θα δημιουργήσει ,όπως και στο IDE, 30 αρχεία μέσα στον φάκελο **Data** τα οποία θα γίνουν read κατευθείαν και θα επιστρέψει το ζητούμενο.

Ο κώδικας λειτουργεί κανονικά και στους 2 και φαίνεται ορθός. Για οποιοδήποτε πρόβλημα αν είναι δυνατόν παρακαλώ επικοινωνήστε μαζί μας. Τα email μας είναι: **p3200278@aueb.gr**, **p3200268@aueb.gr**