Εργασία 2

<u>Συντελεστές</u>

Καραγιάννης Γεώργιος (3190074)

Τάτας Αλέξανδρος (3190196)

Ερώτημα a

Αρχικά ο greedy αλγόριθμος αποτελείται από δυο μεθόδους, την main απ' όπου και τρέχει το πρόγραμμα καθώς και την createTaskArray η οποία καλείται βοηθητικά απ' την main άλλης κλάσης για να δημιουργήσει πίνακα διεργασιών. Η κλάση main δέχεται ως όρισμα ένα αρχείο txt και μέσω των κλάσεων Scanner και Files αρχίζει να το διαβάζει και να το επεξεργάζεται. Διαβάζοντας τον αριθμό των επεξεργαστών αρχίζουμε να του κάνουμε insert στην maxPQ μας ούτως ώστε αργότερα να παίρνουμε κάθε φορά τον επεξεργαστή με την μέγιστη προτεραιότητα (για να του αναθέσουμε διεργασίες) .Ύστερα, διαβάζεται το πλήθος των διεργασιών και διαπιστώνεται εάν υπάρχουν όντως τόσες διεργασίες μέσα στο αρχείο. Τώρα, στο βασικό κομμάτι της χειραγώγησης των δεδομένων έχουμε ως εξής: Από κάθε γραμμή που περιγράφει διεργασία (από την 3η και μετά) παίρνουμε τον πρώτο αριθμό ως id και τον δεύτερο ως χρόνο διεκπεραίωσης και κατασκευάζουμε αντικείμενα τύπου Task. Κάθε φορά που κατασκευάζουμε διεργασία, μέσω την addCompletedTask (μέθοδος της Processor) προσθέτουμε στην λίστα με τις εκπληρωμένες διεργασίες του λιγότερο φορτωμένου χρονικά επεξεργαστή που βρίσκεται εκείνη την στιγμή στην maxPQ. Να σημειωθεί ότι κάθε φορά που επιλέγουμε επεξεργαστή τον αφαιρούμε από την PQ με την μέθοδο get ώστε να τον επεξεργαστούμε και μετά τον κάνουμε πάλι insert για να καταλήξει στην σωστή θέση του σωρού. Είναι προφανές ότι ο πιο φορτωμένος επεξεργαστής είναι και αυτός που καθορίζει ποιο θα είναι το makespan, οπότε κατά την διάρκεια όλων των παραπάνω φροντίζουμε να βρούμε το μεγαλύτερο active time που υπάρχει στον σωρό με τους επεξεργαστές. Τέλος λοιπόν, τυπώνουμε το makespan και αφαιρώντας κάθε επεξεργαστή που βρίσκεται στον σωρό τυπώνουμε αναλυτικά σε μια γραμμή τα στοιχεία του (id, active time και τον χρόνο διεκπεραίωσης κάθε διεργασίας που έχει φέρει εις πέρας).

Ερώτημα b

Επιλέξαμε να υλοποιήσουμε τον αλγόριθμο heapsort και μάλιστα με την μέθοδο του πίνακα. Η κλάση Sort στην οποία υλοποιείται η heapsort αποτελείται από 3 μεθόδους που αλληλεξαρτώνται (heapSort(), heapify(), swap()). Αναλυτικότερα, η swap() λειτουργεί υποστηρικτικά για τις άλλες δυο και ρόλος της είναι να αντιμεταθέτει δυο στοιχεία όποτε της ζητηθεί . Η heapify είναι υπεύθυνη για την κατασκευή του σωρού σε δομή πίνακα. Τέλος, η heapSort() που είναι και η βασική μέθοδος της κλάσης δέχεται ως όρισμα έναν πίνακα διεργασιών και συνδυάζει τις δυο παραπάνω μεθόδους ώστε να επιστρέψει ταξινομημένο τον πίνακα.

Ερώτημα c

Από την υλοποίηση του μέρους Δ σαν συμπέρασμα προκύπτει ότι ο αλγόριθμος greedy decreasing προσφέρει καλύτερη λύση από τον απλό greedy λόγω της ταξινόμησης των διεργασιών. Παρακάτω ακολουθούν τα δεδομένα που προέκυψαν από ένα τρέξιμο του προγράμματος (με δεδομένα εισόδου τα αρχεία στον φάκελο data):

	Greedy	**Greedy decreasing**
N1_1	549	505
N1_2	592	557
N1_3	563	514
N1_4	621	588
N1_5	511	469
N1_6	473	430
N1_7	553	518
N1_8	604	547
N1_9	546	494
N1_10	536	505
N2_1	877	834
N2_2	874	848
N2_3	903	863
N2_4	851	808
N2_5	857	812
N2_6	882	832

N2_7	796	745
N2_8	892	835
N2_9	894	850
N2_10	876	840
N3_1	1224	1180
N3_2	1209	1172
N3_3	1114	1067
N3_4	1145	1109
N3_5	1153	1112
N3_6	1144	1114
N3_7	1147	1098
N3_8	1166	1117
N3_9	1181	1140
N3_10	1180	1119
ΜΕΣΟΣ ΟΡΟΣ	864	820

Ερώτημα d

Στον αλγόριθμο 1 (Greedy) το τρέξιμο του προγράμματος γίνεται μέσω γραμμής εντολών με την καταχώρηση ως όρισμα το μονοπάτι του αρχείου εισόδου. Ένα ενδεικτικό αρχείο στο οποίο βασιστήκαμε για να κάνουμε δοκιμές είναι το test.txt Που βρίσκεται στον φάκελο data της εργασίας που παραδώσαμε.

***Να σημειωθεί ότι λόγω ενός τεχνικού θέματος που δεν καταφέραμε να λύσουμε αντί να καλέσουμε την main της greedy στο πρόγραμμα της comparisons αντιγράψαμε την main μέσα στην ίδια την comparisons αλλάζοντας εννοείται το όνομα της.**