

# ΤΕΧΝΗΤΗ ΝΟΥΜΟΣΗΝΗ

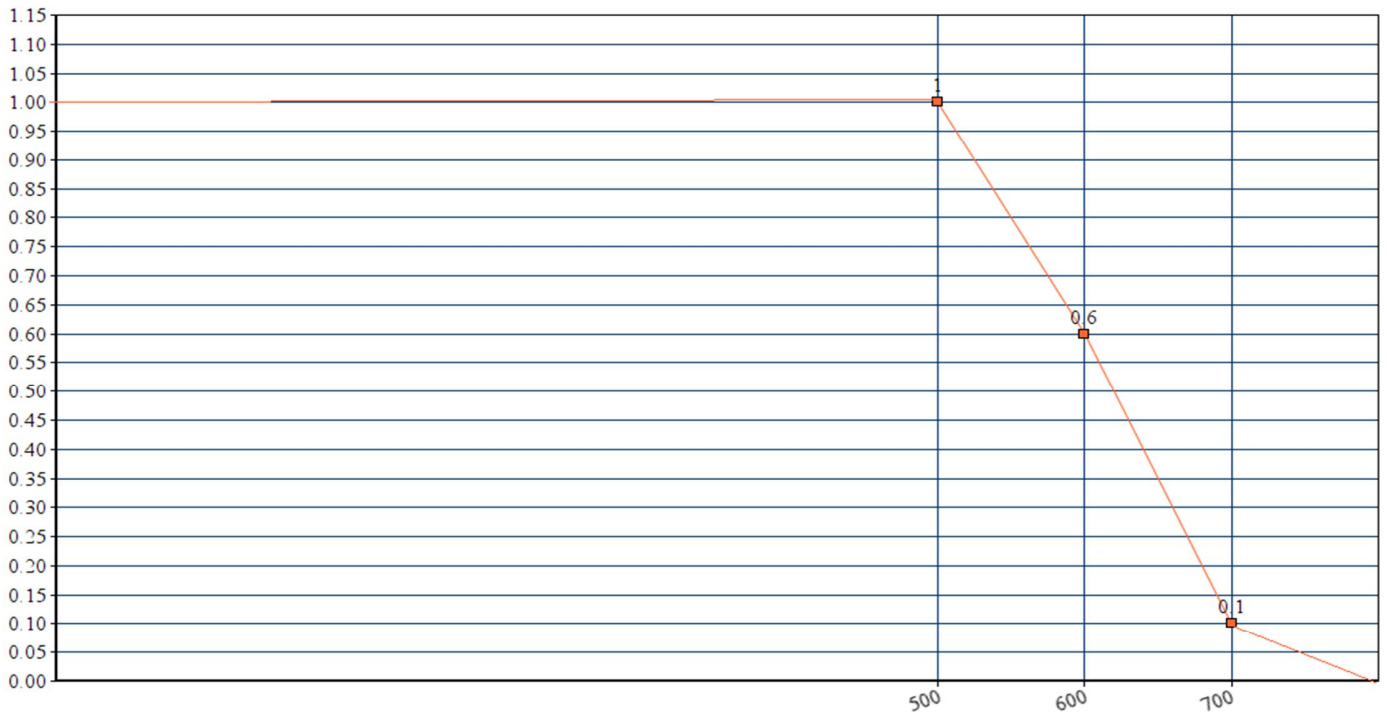
## ΕΡΓΑΣΙΑ 2

DAI16067

ΓΕΩΡΓΙΟΣ ΜΙΧΟΥΛΗΣ

Θα ξεκινήσω περιγράφοντας την όλη διαδικασία που ακολούθησα για να φτάσω τελικά στα διαγράμματα που φαίνονται παρακάτω. Αρχικά άνοιξα το αρχείο `bbsp_generate` με το `codeblocks` και έβαλα τιμές για  $M=12000$  και  $N=20$  έκανα `built` και στην συνέχεια άνοιξα ένα `terminal` με το οποίο έκανα εκτέλεση το αρχείο `.exe` που δημιουργήθηκε από το `bbsp_generate`, (1) `bbsp_generate.exe problem 1 10` και αυτά τα πρώτα 10 προβλήματα που δημιουργήθηκαν ήταν τα μέτριας δυσκολίας προβλήματα. Στην συνέχεια εγώ επέλεξα να τα τρέξω με τον `depth` αλγόριθμο οπότε και ακολούθησα την εξής διαδικασία, άνοιξα το `bbsp` και έκανα `built` το αρχείο, στο `terminal` έτρεξα την εξής εντολή (2) `bbsp.exe depth problem_1.txt solution1d.txt` οπου και δημιουργήθηκε η αντίστοιχη λύση του προβλήματος, το ίδιο έτρεξα (με εναλλαγή στο `solution` τον αριθμό `1d,2d...` και εναλλαγή στο `problem _1 _2...`) και για τα 10 προβλήματα που δημιούργησα πιο πριν και μόλις τα 6 στα 10 είχαν επιτυχία (1,2,4,5,7,9). Υστέρα ήθελα να φτιάξω τα δύσκολης κατηγορίας προβλήματα τα οποία να έχουν στο ανώτερο όριο το κρίσιμο σημείο, για αυτό άνοιξα πάλι το `bbsp_generate.c` και άλλαξα μόνο το  $M$ , οπότε μετα από μερικές δοκιμές παρατήρησα ότι για  $M=14000$  και  $N=20$  είχα το αποτέλεσμα που ήθελα δηλαδή, έκανα `built` το `bbsp_generate` πάλι μετα πήγα στην γραμμή εντολών και έτρεξα την (1) εντολή με νούμερα 11 20 και στην συνέχεια έτρεξα την (2) εντολή με τον ίδιο τρόπο που περιέγραψα παραπάνω και παρατήρησα ότι μόνο 1 στις 10 ήταν επιλυσημη ( η 8<sup>η</sup>) οπότε θεώρησα ότι το κρίσιμο σημείο είναι αυτό γιατί μετα την 1/10 επιτυχίες είναι οι 0/10 επιτυχίες όπου μπορεί να ισχύει για άπειρα μεγαλύτερα  $M$  και ίδιο  $K$  και  $N$ . Αφού δημιούργησα το αρχείο `solution18d.txt` έπρεπε να φτιάξω τώρα και την εύκολη κατηγορία προβλημάτων ώστε όλα να είναι επιλυσημη, στην συνέχεια έκανα πάλι `built` to `bbsp_generate.c` αρχείο με  $M=10000$  ακολούθησα την (1) εντολή με αριθμούς 21 έως 30 και την (2) εντολή με εναλλαγή στα νούμερα από 21 έως 30( `problem_21.txt solution21d.txt`, `problem_22.txt solution22d.txt` ...). Αφού λοιπόν δημιούργησα όλα τα προβλήματα μαζί με τις λύσεις τους

έπρεπε τώρα να κάνω το διάγραμμα επιτυχιών με τον λόγο m/n οπότε μέτρησα σε κάθε λόγο πόσες επιτυχίες είχα και έβγαλα το εξής συμπέρασμα :

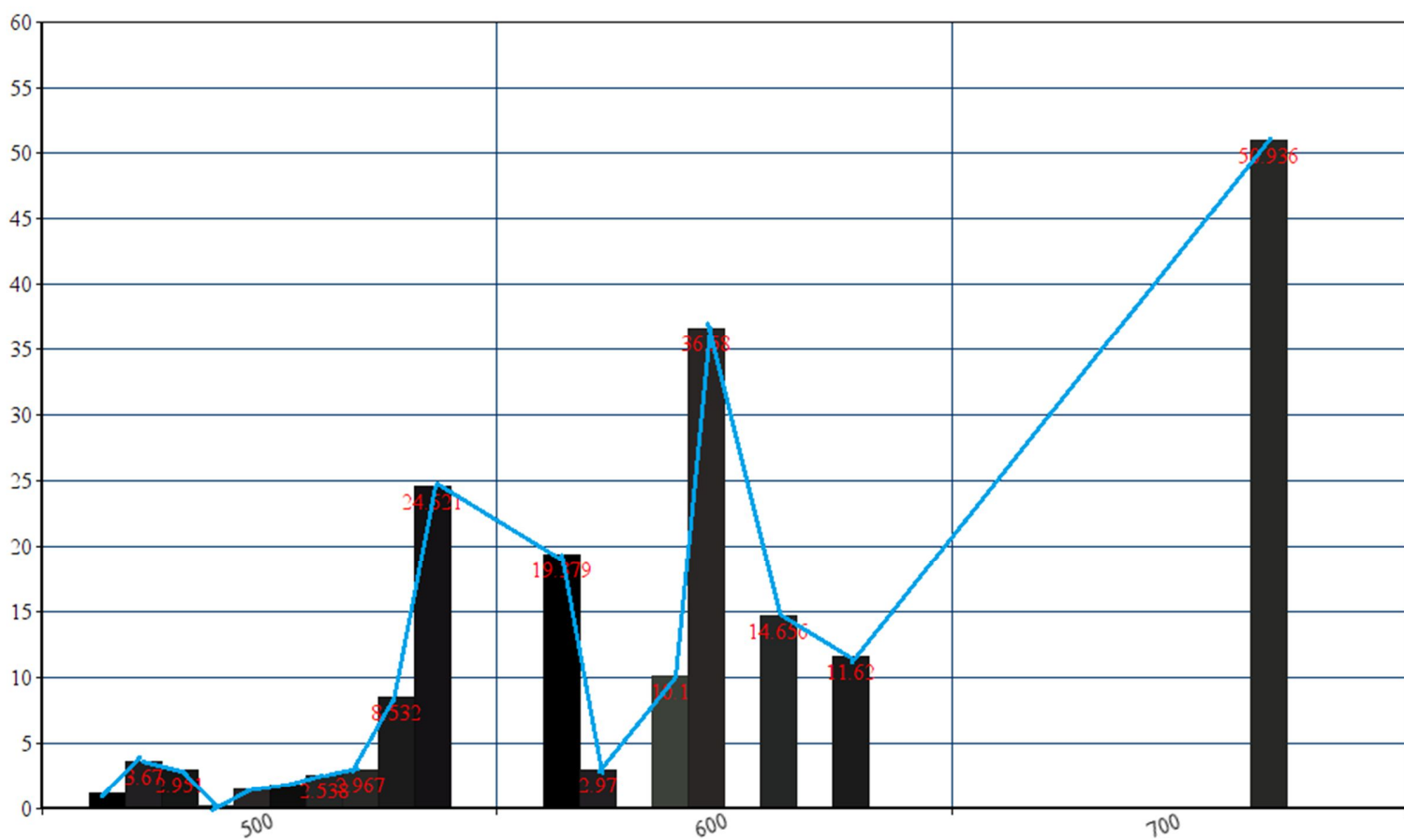


Άξονας Y ποσοστό επιτυχίας, άξονας X λόγος M/N.

Με λόγο 500 (10000/20) όλα είχαν επιτυχία με depth με λόγο 600(12000/20) μόλις το 60% και με λόγο 700 (14000/20) μόλις το 0.1% ,μετα το 0.1 υπάρχει μόνο το 0.

Τώρα όσον αφορά το 2<sup>ο</sup> διάγραμμα εγώ ήδη κάθε φορά που έτρεχα τον αλγόριθμο του depth για το πρώτο ερώτημα τα αποτελέσματα τα αποθήκευα σε ένα αρχείο με όνομα Depth.txt εκεί μέσα είχα τα αποτελέσματα του terminal, οπότε πήρα τους χρόνους και έφτιαξα ένα διάγραμμα μόνο για το depth : στον άξονα y έχουμε τον χρόνο σε

δευτερόλεπτα και στον άξονα χ τον λόγο M/N και αφορά μόνο τα επιλυσημα.

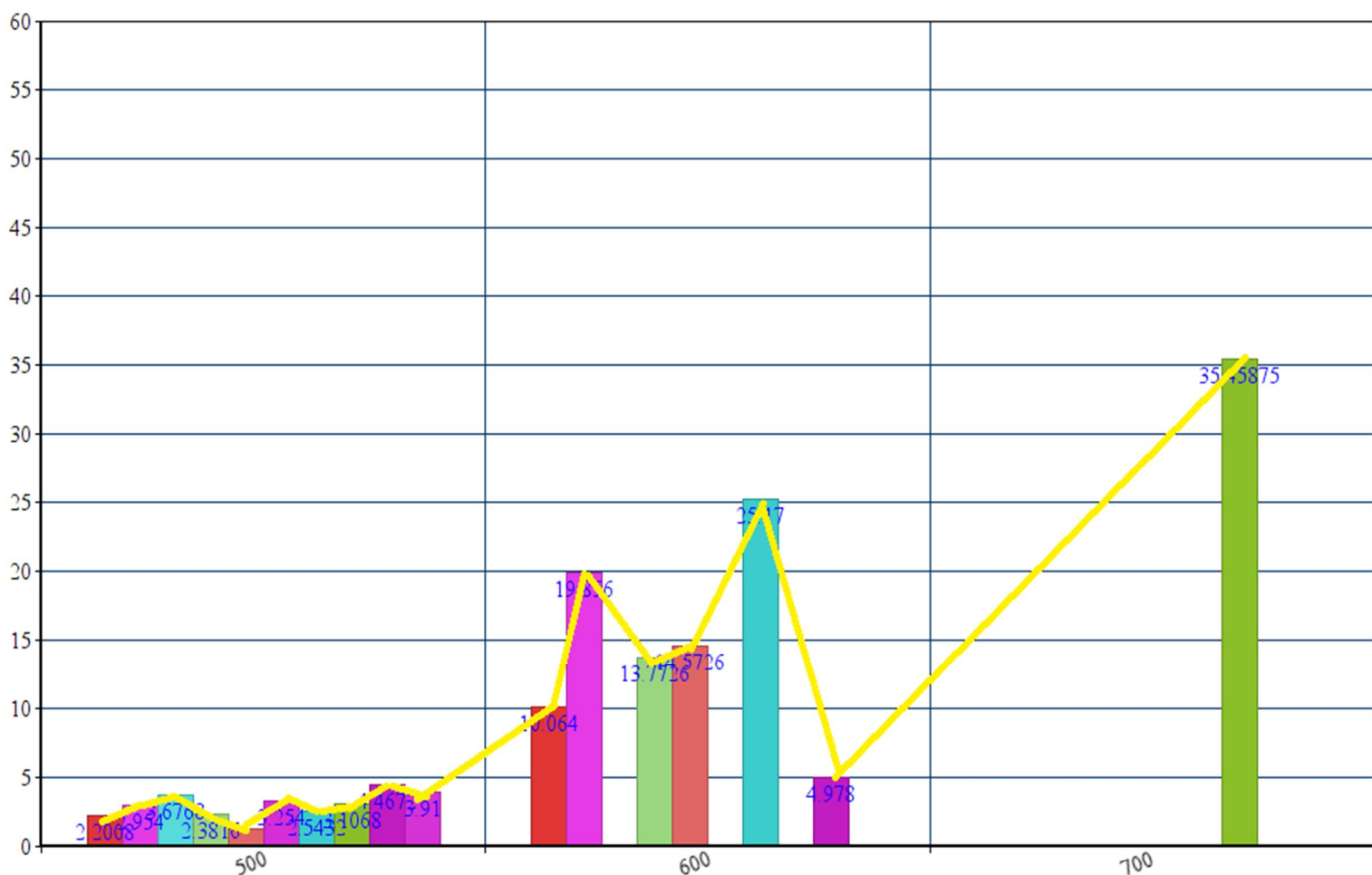


Τώρα όμως έπρεπε να βρω και τους χρόνους για τον hill οπότε όπως έγραφαν οι οδηγίες έσπερε να τρέξω τον αλγόριθμο του hill 5 φορές για

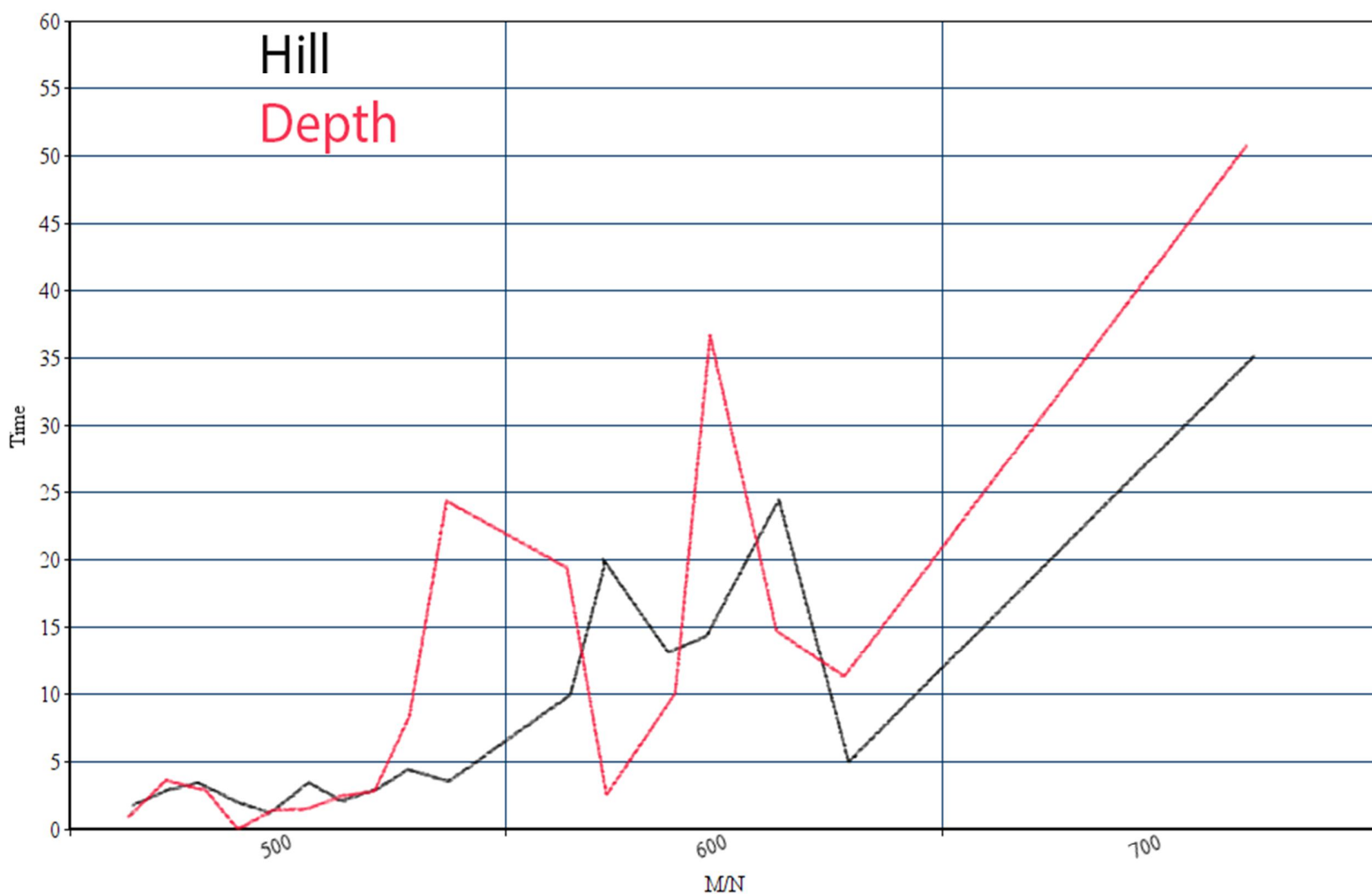
το κάθε πρόβλημα και έτσι και έκανα, στο terminal έτρεξα την εντολή  
bcsp.exe hill problem\_1.txt solution11h.txt

Το 11 h σημαίνει να δημιουργήσει την 1<sup>η</sup> λύση για το 1<sup>ο</sup> πρόβλημα οπότε ακολουθήσε αυτή η αρίθμηση 11h ,12h...( έως 9 το πρώτο ψηφίο και έως 5 το 2<sup>ο</sup>) στο πρόβλημα του 11 είχα την αρίθμηση 111 h 112h... και στο 21 211h ,212h.... (έως 5 το τελευταίο ψηφίο και έως 9 το μεσαίο) . Τα αποτελέσματα του hill τα αποθήκευσα σε ένα αρχείο Hill.txt μ μέσα σε αυτό το αρχείο έχω μόνο τα επιλυσημα προβλήματα του depth αντίστοιχα ( δηλαδή αν το depth έλυσε το 20<sup>ο</sup> πρόβλημα τότε το έτρεξα και με hill αν όχι δεν το έτρεξα με hill ακόμα και άμα το έλυne ο hill). Οπότε από τις 5 φορές που έτρεξα το κάθε πρόβλημα με τον αλγόριθμο του hill πήρα τον μέσο ορό του χρόνου τους , αν τώρα ο hill κατάφερne και έλυne μόνο 1 φορά το πρόβλημα στις 5 προσπάθειες τότε έπαιρνα αυτόν τον χρόνο σαν δεδομένο ( πρόβλημα 18). Με αυτόν τον τρόπο έφτιαξα και το διάγραμμα χρόνου(άξονας Y) με m/n(άξονας X):

Οι τιμές που αναγράφονται πάνω στις ράβδους είναι οι μεσοί όροι.



Τώρα έπρεπε να τα συνδυάσω αυτά τα δυο διαγράμματα ώστε να μπορέσω να διακρίνω τις διαφορές των 2 αλγορίθμων οπότε και έτσι έκανα, χρησιμοποίησα το πρόγραμμα photoshop και ένωσα τα 2 ιστογράμματα σε 1.



Ευκολά κάνεις παρατηρεί ότι ο αλγόριθμος του Hill είναι καλύτερος χρονικά. Όσον αφορά τα διαγράμματα δυστυχώς προσπαθούσα να το κάνω με διάφορους τρόπους όμως επειδή όλες οι τιμές είχαν 3 διαφορετικά  $x$  και πολλά διαφορετικά  $y$  είχαν κολλήσει το ένα πάνω στο άλλο και δεν φαινότουσαν, οπότε έκανα ραβδογραμματα τα οποία στοιχήθηκαν έτσι με τέτοια σειρά με βάση την σειρά των προβλημάτων (πχ πρώτη τιμή είναι από το πρόβλημα 1 ,2<sup>η</sup> τιμή από το πρόβλημα 2...) και μόνο έτσι κατάφερα να μην είναι κολλημένες οι τιμές η μια πάνω στην

άλλη αλλά όλα έχουν τον ίδιο λόγο  $M/N$  και στην συνέχεια με το πρόγραμμα photoshop ένωσα το πάνω κέντρο τους και δημιούργησα τα ιστογράμματα.

Στον φάκελο Source code περιέχονται όλα τα αρχεία που δημιούργησα κατά την διάρκεια της άσκησης.