



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Δ.Π.Μ.Σ. ΕΠΙΣΤΗΜΗΣ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Προγραμματιστικά Εργαλεία και Τεχνολογίες  
για την Επιστήμη Δεδομένων

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ ΣΤΗ ΡΥΤΗΘΝ

ΤΟΥ

ΜΑΡΙΟΥ ΒΛΑΧΟΓΙΑΝΝΟΠΟΥΛΟΥ

03400205

Διδάσκων: Αθανάσιος Βουλόδημος

Επ. Καθηγητής Ε.Μ.Π.

## Περιεχόμενα

1	Το πρόβλημα time-travel	3
2	Περιγραφή βασικών στοιχείων της λύσης	4
3	Περίπτωση small.txt	8
4	Περίπτωση large.txt	10
5	Επίλογος	10
	Βιβλιογραφία	12

## Κατάλογος Σχημάτων

2.1	Εξέλιξη cscs με τη πρόθεση για αγορά ή για πώληση . . . . .	5
2.2	Εξέλιξη amzn με τη πρόθεση για αγορά ή για πώληση . . . . .	6
3.1	Επικύρωση ακολουθίας small.txt . . . . .	8
3.2	Εξέλιξη υπολοίπου small.txt . . . . .	9
3.3	Εξέλιξη αποτίμησης μετοχών small.txt . . . . .	9
4.1	Επικύρωση ακολουθίας large.txt . . . . .	10
4.2	Εξέλιξη υπολοίπου large.txt . . . . .	11
4.3	Εξέλιξη αποτίμησης μετοχών large.txt . . . . .	11

## Κατάλογος Αλγορίθμων

2.1	Γενική μέθοδος trade . . . . .	7
-----	--------------------------------	---

# 1 Το πρόβλημα time-travel

Στο πρόβλημα time-travel έχουμε στη διάθεση μας ένα σύνολο δεδομένων [1] με τις ιστορικές τιμές όλων των μετοχών του χρηματιστηρίου της Νέας Υόρκης. Σκοπός μας είναι να βρούμε ακολουθίες από αγοραπωλησίες ώστε να συγκεντρώσουμε πολλά χρήματα στο τέλος του χρονικού ορίζοντα 11-10-2017, αν ξεκινάμε από 1\$ στις 01-01-1960.

Συγκεκριμένα, ο κατάλογος Stocks του dataset περιέχει για κάθε μετοχή  $s$  ένα αρχείο CSV με όνομα  $s.us.txt$  — για παράδειγμα, για τη μετοχή AAPL της Apple Inc. υπάρχει το αρχείο με όνομα  $aapl.us.txt$ . Κάθε τέτοιο αρχείο περιέχει, μεταξύ άλλων, για κάθε ημερομηνία  $d$  την τιμή ανοίγματος (open) της μετοχής  $s$  τη μέρα  $d$ , τη μέγιστη τιμή (high) της μετοχής  $s$  τη μέρα  $d$ , την ελάχιστη τιμή (low) της μετοχής  $s$  τη μέρα  $d$ , την τιμή κλεισίματος (close) της μετοχής  $s$  τη μέρα  $d$  και τον όγκο (volume) των αγοραπωλησιών της μετοχής  $s$  τη μέρα  $d$ .

Σκοπός μας είναι η εύρεση μίας ακολουθία  $N$  κινήσεων, κάθε μία από τις οποίες θα είναι μία τετράδα της μορφής  $(d_i, m_i, s_i, x_i)$  για  $0 \leq i < N$ , όπου  $d_i$  είναι η μέρα κατά την οποία θα εκτελεστεί,  $m_i$  ο κωδικός της κίνησης,  $s_i$  ο κωδικός της μετοχής που αφορά και  $x_i$  το πλήθος των μετοχών είτε προς αγορά είτε προς πώληση.

Τα βασικά είδη κινήσεων που μας συμφέρει να εκτελούμε είναι τα εξής:

- buy-low: αγόρασε στην ελάχιστη τιμή.
- sell-high: πούλησε στη μέγιστη τιμή.

Βάσει των πληροφοριών που έχουμε στη διάθεσή μας, γνωρίζουμε την ελάχιστη και τη μέγιστη τιμή κάθε μετοχής μέσα στη μέρα αλλά δε γνωρίζουμε με ασφάλεια αν η ελάχιστη τιμή προηγήθηκε χρονικά της μέγιστης ή αντίστροφα. Για το λόγο αυτό, θεωρούμε ότι αυτές οι αγορές και οι πωλήσεις γίνονται ανεξάρτητα, δηλαδή πρέπει εκ των προτέρων να έχουμε διαθέσιμα χρήματα για τις μετοχές που θέλουμε να αγοράσουμε και επίσης να έχουμε στην κατοχή μας τις μετοχές που θέλουμε να πουλήσουμε.

Ακόμη, μπορούμε να αγοράζουμε και πουλάμε την ίδια μέρα, εκμεταλλευόμενοι τη διακύμανση των μετοχών (intra-day trading). Για το λόγο αυτό, προσθέτουμε και τα εξής είδη κινήσεων:

- buy-open: αγόρασε στην τιμή ανοίγματος.
- sell-open: πούλησε στην τιμή ανοίγματος.
- buy-close: αγόρασε στην τιμή κλεισίματος.
- sell-close: πούλησε στην τιμή κλεισίματος.

Οι κινήσεις εκτελούνται με χρονολογική σειρά. Μέσα στην ίδια μέρα, οι κινήσεις εκτελούνται με την εξής σειρά:  $\{\text{buy-open}, \text{sell-open}\} > \{\text{buy-low}, \text{sell-high}\} > \{\text{buy-close}, \text{sell-close}\}$  δηλαδή σε τρεις ομάδες: κατά το άνοιγμα, κατά τη διάρκεια της μέρας και κατά το κλείσιμο. Πριν να εκτελεστούν οι εντολές κάθε ομάδας, πρέπει να υπάρχουν στο λογαριασμό αρκετά χρήματα για όλες τις εντολές αγορών και να έχουμε στην κατοχή μας όλες τις μετοχές προς πώληση. Η σειρά με την οποία εκτελούνται οι εντολές κάθε ομάδας δεν έχει σημασία.

Ενας περιορισμός που τίθεται είναι το συνολικό πλήθος των μετοχών  $s$  που αγοράζουμε όσο και το συνολικό πλήθος των μετοχών  $s$  που πουλάμε την ημέρα  $d$  να μην ξεπερνά το 10% του όγκου της μετοχής  $s$  την ημέρα  $d$ .

Επισυναπτόμενα θα βρείτε, εκτός από το παρόν:

- small.txt: αρχείο με  $N \leq 1000$  κινήσεις.
- large.txt: αρχείο με  $N \leq 1000000$  κινήσεις.
- stocks-project.zip: zip με το κώδικα:
  - /src: directory με το κώδικα:
    - \* files.py: python module με συναρτήσεις για το διάβασμα αρχείων του dataset και τη δημιουργία των αρχείων small.txt, large.txt
    - \* portfolio.py: python module με τη κλάση Portfolio για τη διαχείριση του λογαριασμού (μετοχές, υπόλοιπο)
    - \* trade.py: python module με τη κλάση Trader για την έκδοση εντολών αγοραπωλησίας
    - \* Stocks.ipynb: notebook με τη προεπεξεργασία του συνόλου δεδομένων, σχο- λιασμούς, γραφικές παραστάσεις κ.α.
  - /Stocks: άδειο directory που μπαίνουν τα .us.txt
  - pyproject.toml: αρχείο με τα dependencies
  - poetry.toml: για να μη ψάχνετε το .venv

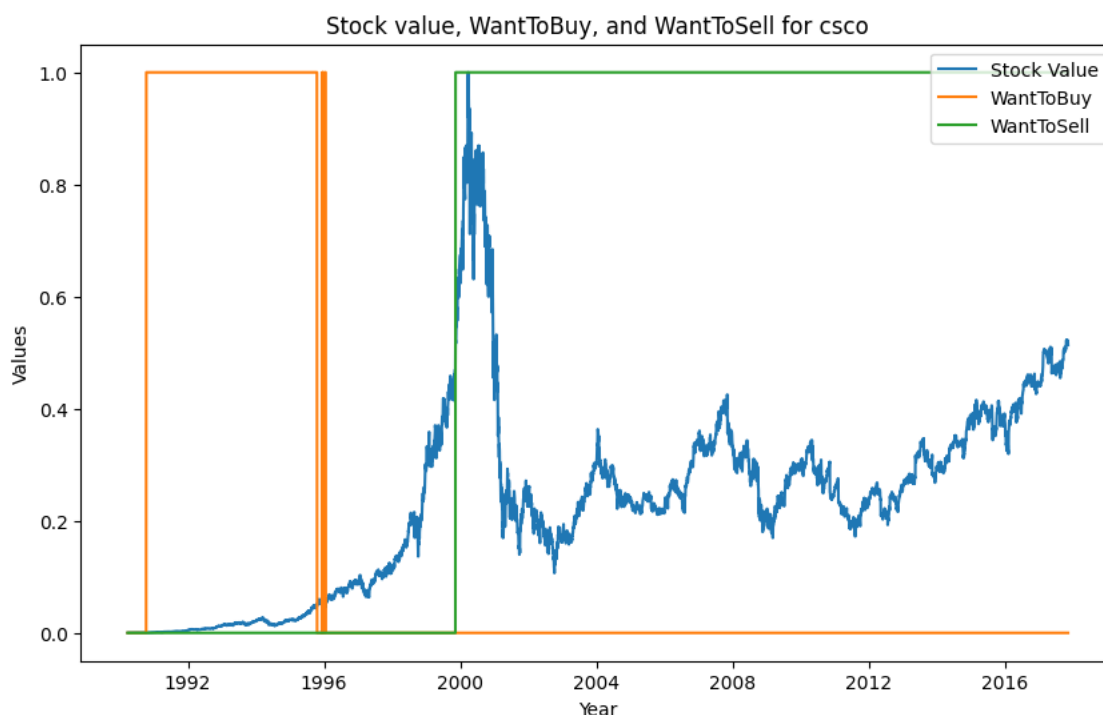
Προτείνεται η χρήση poetry [2] για τη διαχείριση των dependencies. Με poetry install παράγεται το .venv. Επιλέξτε την Python εντός του .venv ως interpreter. Στη χειρότερη κάντε copy-paste τις συναρτήσεις και κλάσεις από το /src σε ένα notebook στο Colab.

## 2 Περιγραφή βασικών στοιχείων της λύσης

Η βασική ιδέα της λύσης είναι η συσσώρευση μεγάλου πλήθους μετοχών όταν έχουν χαμηλή αξία, η αναμονή μέχρι η τιμή τους να ανέβει και η πώληση τους όταν η τιμή τους είναι υψηλή. Στο ενδιάμεσο, εκτελούμε intraday trading για να βγάλουμε χρήματα για να αγοράσουμε και άλλες μετοχές.

Το διάβασμα των αρχείων με τα δεδομένα των μετοχών, έχει ως αποτέλεσμα ένα μεγάλο dataframe με μια γραμμή για κάθε δυάδα (μετοχής, μέρας), εφεξής μετοχομέρα, και στήλες τις τιμές Open, High, Low, Close και Volume. Οι στήλες Date, Name καθορίζουν τη μοναδικότητα, και τίθενται ως index για γρήγορες αναζητήσεις.

Γενικά επιλέγουμε τις μετοχές με τη μεγαλύτερη ποσοστιαία αύξηση τιμής κλεισίματος από ελάχιστο στο μέγιστο στη διάρκεια της ζωής τους και οι οποίες εμφανίζουν την ελάχιστη τιμή κλεισίματος χρονικά πριν από τη μέγιστη (θα τις λέμε stonks). Έτσι, αφού φιλτράρουμε τις μετοχές για τις οποίες  $DateOfMinClose < DateOfMaxClose$ , τις ταξινομήσουμε σε φθίνουσα σειρά με βάση τη  $(max(close) - min(close))/min(close)$ , και από αυτές κρατάμε τις πρώτες 50. Σε αυτές τις μετοχές, συμπεριλαμβάνουμε και τη ge καθώς είναι μετοχή με



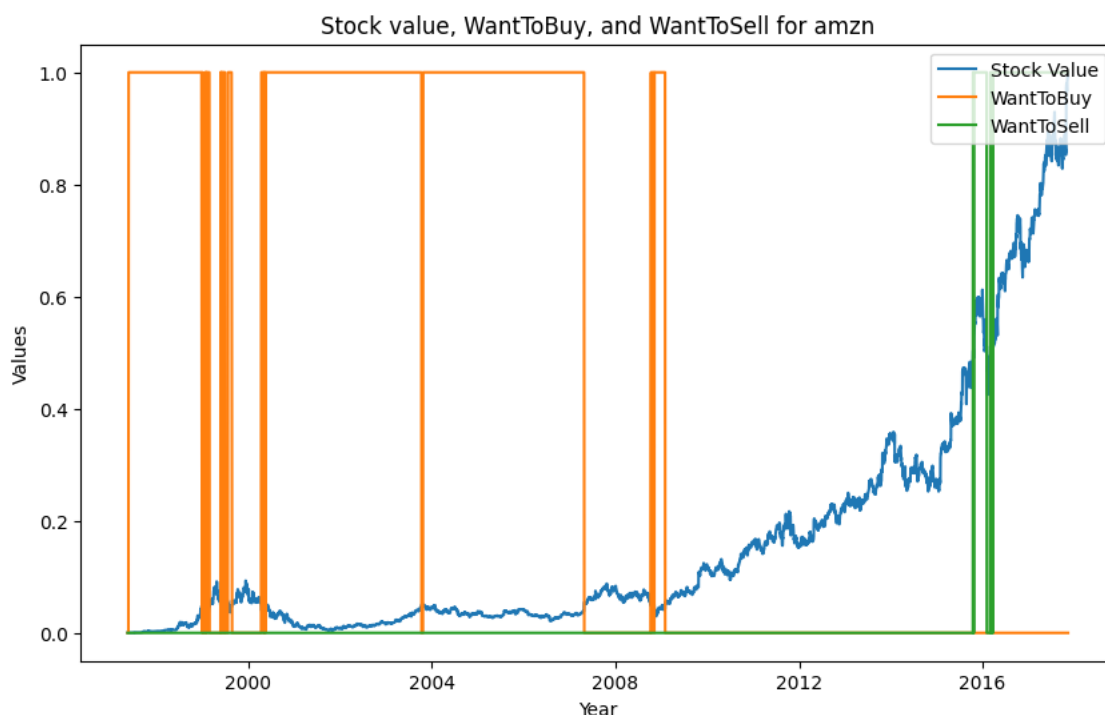
Σχήμα 2.1: Εξέλιξη cscs με τη πρόθεση για αγορά ή για πώληση

μεγάλη αύξηση (όχι στις top 50 όμως) μέχρι τις αρχές του 2000 και επίσης είναι η πρώτη εταιρία του dataset με δεδομένα από τις 01-01-1960. Σημειώνεται ότι προηγείται γκρουπάρισμα του συνόλου των μετοχοημερών με βάση τη μετοχή.

Φιλτράρουμε το dataframe με τις μετοχομέρες ώστε να περιέχει μόνο τις stonk μετοχές. Για κάθε μετοχομέρα υπολογίζουμε το ποσοστό αύξησης της μετοχής τη συγκεκριμένη μέρα από την συνολικά ελάχιστη τιμή της:  $PercentageIncrease = (Close - \min(close)) / (\max(close) - \min(close))$ . Με βάση αυτό το μέτρο καθορίζουμε το κατά πόσο θέλουμε να αγοράσουμε τη συγκεκριμένη μετοχή τη συγκεκριμένη μέρα ή να τη πουλήσουμε. Συγκεκριμένα, θέλουμε να αγοράζουμε αυτή τη μετοχή τις μέρες για τις οποίες  $PercentageIncrease < 0.05$  και να τη πουλάμε όταν  $PercentageIncrease > 0.5$ . Τα κατώφλια αυτά προέκυψαν εμπειρικά. Προφανώς τις μέρες που προηγούνται της μέρας της ελάχιστης τιμής κλεισίματος δεν θέλουμε να αγοράζουμε ή να πουλάμε, ενώ τις μέρες που έπονται τη μέρας με τη μέγιστη τιμή κλεισίματος θέλουμε μόνο να πουλάμε. Στα σχήματα 2.1, 2.2 φαίνεται το μέτρο  $PercentageIncrease$  καθώς και η εξέλιξη των μέτρων  $WantToBuy$ ,  $WantToSell$  για τις μετοχές cscs, amzn αντίστοιχα.

Γενικά όλες οι μετοχές που έχουμε επιλέξει έχουν παρόμοιες ραγδαίες αυξήσεις στη τιμή τους. Κάποιες, όπως η amzn συνεχώς αυξάνονται μέχρι το τέλος του χρονικού ορίζοντα, ενώ άλλες, όπως τη cscs, αυξάνονται ραγδαία μέχρι ένα σημείο και μετά ξεκινάνε να πέφτουν. Τα μέτρα  $WantToBuy$ ,  $WantToSell$  εκμεταλλεύονται αυτό το γεγονός, ώστε να παρατάμε τη συσσώρευση μιας μετοχής και να μεταπηδάμε σε άλλη.

Επειδή, σκοπός μας είναι η συσσώρευση μετοχών, χρησιμοποιούμε intraday trading,



Σχήμα 2.2: Εξέλιξη amzn με τη πρόθεση για αγορά ή για πώληση

μόνο για να συγκεντρώσουμε χρήματα για να αγοράσουμε και άλλες μετοχές. Συγκεκριμένα, εντοπίζουμε για όλες τις μετοχομέρες, αυτές για τις οποίες  $(Open = High \wedge High > Low) \vee (Close = Low \wedge High > Low)$ , με σκοπό να πουλήσουμε όσες μετοχές μας επιτρέπει ο κανόνας του 10% του Volume στο High (Open ή High αντίστοιχα) και να τις ξαναγοράσουμε όλες στο Low (Low ή Close αντίστοιχα). Έτσι, βγάζουμε χρήματα και κρατάμε όλες τις μετοχές μας.

Επειδή, υπάρχει περίπτωση να συσσωρεύσουμε τόσες πολλές μετοχές, που να μην προλαβαίνουμε να τις πουλήσουμε στο τέλος (ειδικά για τις μετοχές που εμφανίζουν το μέγιστο Close κοντά στο τέλος του ορίζοντα), υπολογίζουμε για κάθε μετοχομέρα το πλήθος των μετοχών που μπορούμε να πουλήσουμε για αυτή τη μετοχή από αυτή τη μέρα μέχρι το τέλος του χρονικού ορίζοντα (MaxCanSellUntilEnd). Αν για μια μετοχομέρα, το πλήθος της συγκεκριμένης μετοχής που έχουμε στη διάθεση μας, ξεπερνά το MaxCanSellUntilEnd, τότε τη πουλάμε.

Για να αποφύγουμε, να παραβούμε τον κανόνα του 10% του Volume, δεν μπορούμε σε μια μέρα για μια μετοχή να προβούμε και σε αγορά και σε πώληση και σε intraday trading. Σε κάθε περίπτωση προτεραιότητα παίρνει η πώληση, η οποία σίγουρα δεν συμπίπτει ποτέ με την αγορά σύμφωνα με το τρόπο υπολογισμού των *WantToBuy*, *WantToSell*. Ανάμεσα στο intraday trading και στην αγορά, επιλέγουμε την αγορά στη περίπτωση small.txt, ενώ το intraday trading στη περίπτωση του large.txt.

Επίσης, για τον ίδιο λόγο, μια μετοχή δεν μπορεί να συμμετέχει και στο sell-open - buy-low και στο sell-high - buy-close βήμα του intraday trading.

```

for today, stocks_today in df.groupby("Date"):
    get todays_stocks_i_must_sell # using MaxCanSellUntilEnd

    # these two are always mutually exclusive
    get sell_high_rebuy_close # different method for small.txt/large.txt
    get sell_open_rebuy_low # different method for small.txt/large.txt

    sell_high_rebuy_close &= ~todays_stocks_i_must_sell

    sell_open_rebuy_low &= ~todays_stocks_i_must_sell

    """Opening Phase"""
    save_money = save_some_money_to_rebuy_on_low(sell_open_rebuy_low)
    execute sell_open_rebuy_low # sell as much as we can

    """Middle Phase"""
    money_to_spent_on_buy = balance - save_money
    get stocks_i_want_to_buy # using WantToBuy
    stocks_i_want_to_buy &= (
        (~sell_high_rebuy_close)
        & (~sell_open_rebuy_low)
        & (~todays_stocks_i_must_sell)
    )
    get stocks_i_want_to_sell # using WantToSell
    stocks_i_want_to_sell |= todays_stocks_i_must_sell

    execute stocks_i_want_to_buy # split money_to_spent_on_buy equally
    execute stocks_i_want_to_sell # sell as much as we can
    execute sell_open_rebuy_low # rebuy all my stocks back
    execute sell_high_rebuy_close # sell as much as we can

    """Closing Phase"""
    execute sell_high_buy_close_trades # rebuy all my stocks back

```

### Αλγόριθμος 2.1: Γενική μέθοδος trade

Στο 2.1 παρουσιάζεται σε ψευδο-python η γενική μέθοδος που χρησιμοποιείται. Η υλοποίηση μπορεί να βρεθεί στη μέθοδο trade της κλάσης Trader στο module trade.py. Η διαφοροποίηση μεταξύ small.txt και large.txt, αφορά την επιλογή των σημερινών μετοχών που θα χρησιμοποιηθούν για intraday trading.

### 3 Περίπτωση small.txt

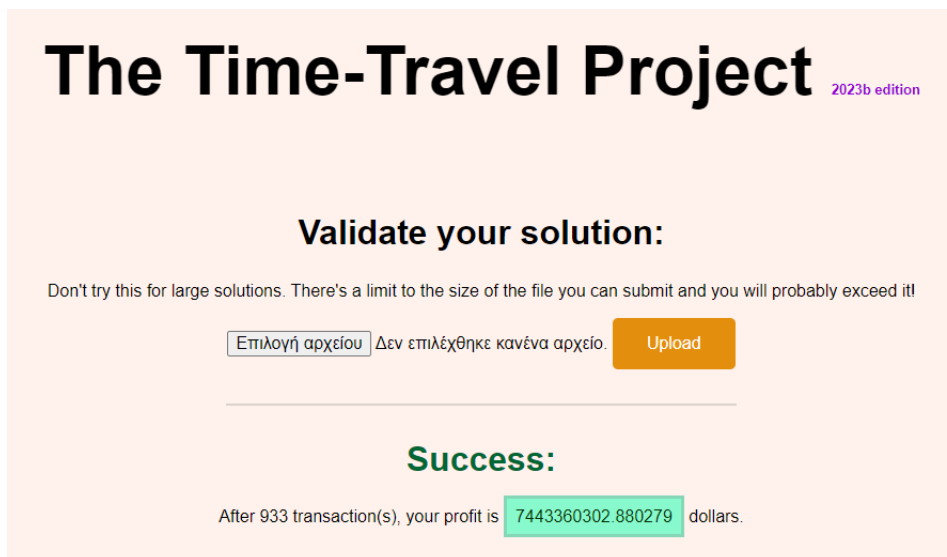
Γενικά, η στρατηγική που αναλύθηκε στο μέρος 2, ευνοείται από μεγάλο πλήθος συναλλαγών, ειδικά intraday trading συναλλαγών, στις οποίες πουλάμε και αγοράζουμε όσο περισσότερο μπορούμε από κάθε είδος μετοχής που διαθέτουμε. Συνεπώς ο περιορισμός του πλήθους των κινήσεων αποτελεί σημαντική πρόκληση.

Αρχικά, επιλέγουμε να εφαρμόσουμε τη μέθοδο για 2 μόνο εταιρίες: τη cscs και τη amzn. Η εξέλιξη των τιμών τους φαίνεται στα σχήματα 2.1 και 2.2 αντίστοιχα.

Για να περιορίσουμε το πλήθος των συναλλαγών, περιορίζουμε τις μέρες που αξίζει να κάνουμε intraday-trading, στις πρώτες 500 με το μεγαλύτερο μέτρο  $(High - Low) * Volume / Low$ . Το μέτρο αυτό ευνοεί τα υψηλά κέρδη intraday-trading, αλλά τιμωρεί όταν το *Low* είναι επίσης υψηλό. Η ιδέα είναι ότι επιλέγουμε να κάνουμε intraday-trading κυρίως στο early game που τα *Low* είναι χαμηλά, ενώ στο late game μας αρκεί η stonk φύση των μετοχών. Τις μέρες που δεν αξίζει να κάνουμε intraday-trading, τις αφήνουμε ως έχουν.

Επίσης, για να περιορίσουμε ακόμα περισσότερο το πλήθος των συναλλαγών, τροποποιούμε τη λογική με την οποία επιλέγουμε να κάνουμε intraday-trading. Συγκεκριμένα, κάνουμε intraday-trading το πολύ μία μετοχή (αυτή που θα δώσει το μεγαλύτερο κέρδος) για το sell open - rebuy low και το πολύ μια μετοχή για το sell high - rebuy close και μόνο όταν δεν έχουμε χρήματα να αγοράσουμε καμία μετοχή από τις WantToBuy μετοχές της μέρας. Δίνουμε δλδ προτεραιότητα στην αγορά, αντί για το intraday-trading, σε αντίθεση με τη large εκδοχή που θα δούμε παρακάτω. Η επιλογή των μετοχών για intraday-trading μπορεί να βρεθεί στη μέθοδο `_small_intraday_opportunity` της κλάσης `Trader` στο module `trade.py`.

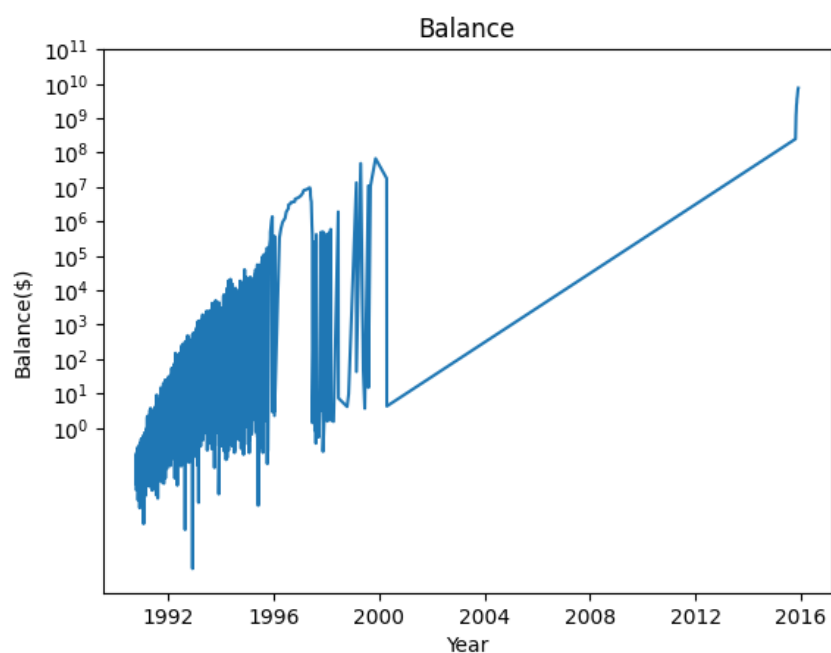
Τρέχοντας τον αλγόριθμο 2.1 για τη small εκδοχή λαμβάνουμε το παραδοθέν small.txt. Το αποτέλεσμα των συναλλαγών είναι να καταλήγουμε με 7443360302.880279\$ (7.4 δις) και 933 συναλλαγές. Το αποτέλεσμα επιβεβαιώνεται από τη ιστοσελίδα του μαθήματος [3], όπως φαίνεται στο σχήμα 3.1



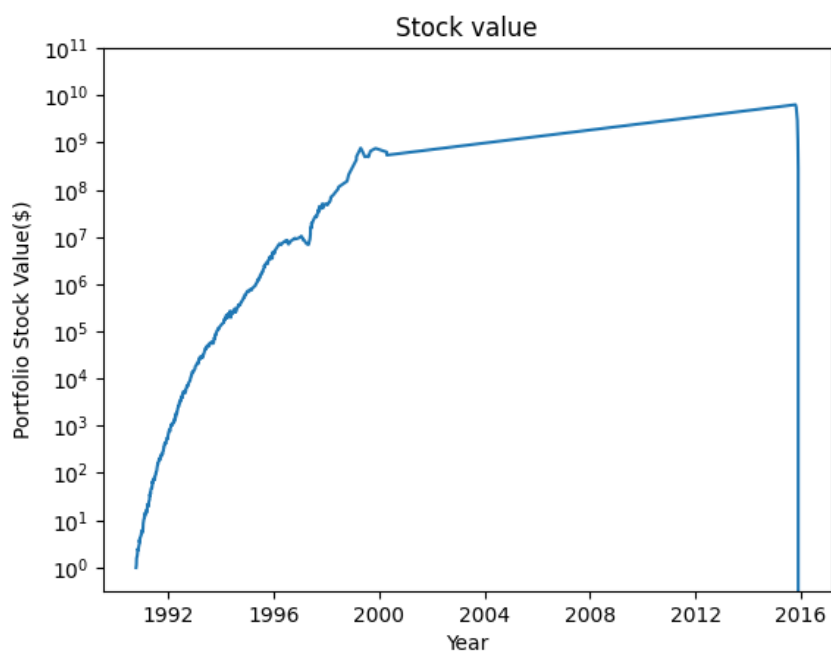
Σχήμα 3.1: Επικύρωση ακολουθίας small.txt



Στα σχήματα 3.2, 3.3 φαίνεται η εξέλιξη του υπολοίπου συναλλαγών και η εξέλιξη αποτίμησης μετοχών που έχουμε στη διάθεσή μας.



Σχήμα 3.2: Εξέλιξη υπολοίπου small.txt

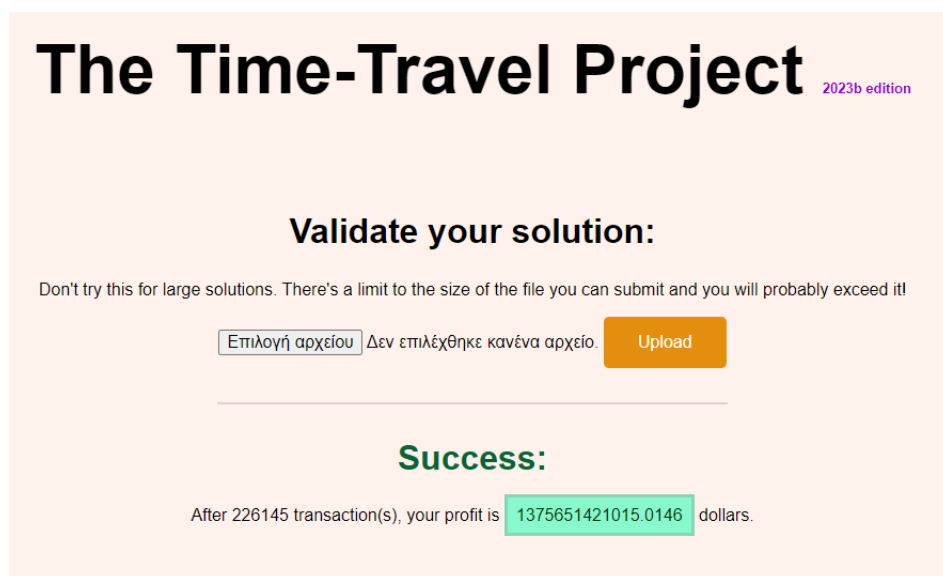


Σχήμα 3.3: Εξέλιξη αποτίμησης μετοχών small.txt

## 4 Περίπτωση large.txt

Για το μεγάλο παράδειγμα χρησιμοποιούμε και τις 51 εταιρείες. Οι τροποποιήσεις, αναφέρθηκαν και στα προηγούμενα, αλλά εν γένει αφορούν την προτίμηση για intraday trading έναντι της αγοράς καθώς και τη πώληση και επαναγορά όλων των μετοχών που έχουμε στη διάθεση μας, όταν κάνουμε intraday trading. Η επιλογή των μετοχών για intraday trading μπορεί να βρεθεί στη μέθοδο `_intraday_opportunity` της κλάσης `Trader` στο module `trade.py`.

Τρέχοντας τον αλγόριθμο 2.1 για τη large εκδοχή λαμβάνουμε το παραδοθέν large.txt. Το αποτέλεσμα των συναλλαγών είναι να καταλήγουμε με 1375651421015.0146\$ (1.37 τρις) και 226145 συναλλαγές. Το αποτέλεσμα επιβεβαιώνεται από τη ιστοσελίδα του μαθήματος [3], όπως φαίνεται στο σχήμα 4.1. Σημειώνεται ότι στο τέλος του χρονικού ορίζοντα μας μένουν και 1150126535.21 (1.1 δις) σε μετοχές που δεν προλάβουμε να πουλήσουμε εγκαίρως.

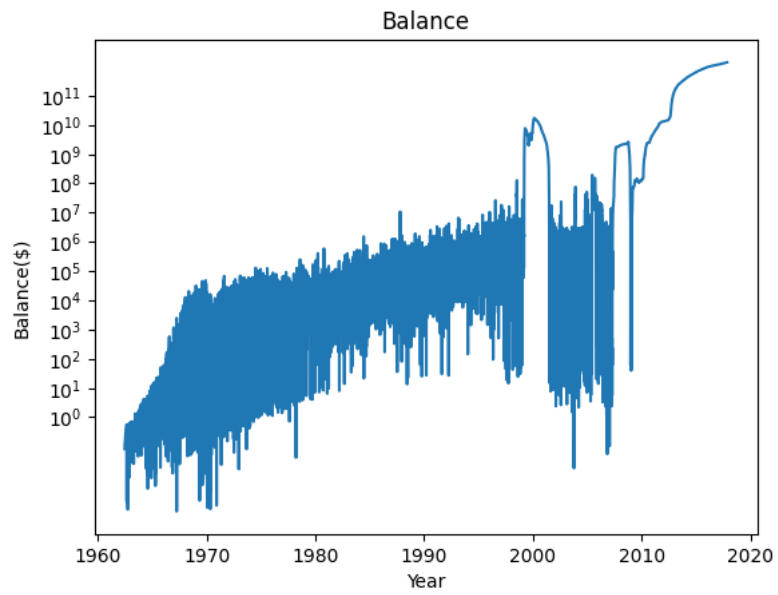


Σχήμα 4.1: Επικύρωση ακολουθίας large.txt

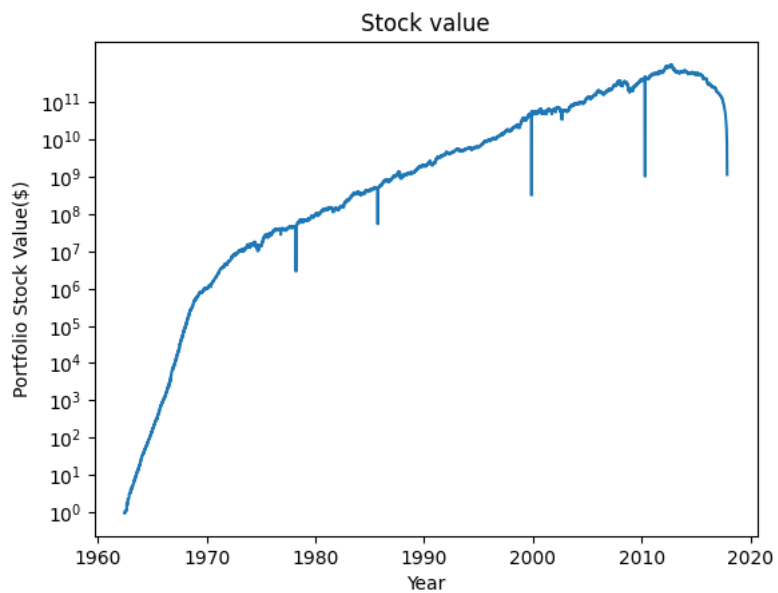
Στα σχήματα 4.2, 4.3 φαίνεται η εξέλιξη του υπολοίπου συναλλαγών και η εξέλιξη αποτίμησης μετοχών που έχουμε στη διάθεσή μας.

## 5 Επίλογος

Σε γενικές γραμμές, η μέθοδος που αναπτύχθηκε πετυχαίνει καλές απολαβές στη large περίπτωση. Παρολαυτά επειδή αναπτύχθηκε με τη large περίπτωση σαν main case, μάλλον απέχει πολύ από τη βέλτιστη λύση στη small περίπτωση. Επίσης μάλλον απέχει πολύ και από τη βέλτιστη λύση στη large περίπτωση, αν και είναι δύσκολο να το διαπιστώσουμε.



Σχήμα 4.2: Εξέλιξη υπολοίπου large.txt



Σχήμα 4.3: Εξέλιξη αποτίμησης μετοχών large.txt

## Βιβλιογραφία

- [1] *Huge Stock Market Dataset*. <https://www.kaggle.com/datasets/borismarjanovic/price-volume-data-for-all-us-stocks-etfs>.
- [2] *Poetry: Python packaging and dependency management*. <https://python-poetry.org/docs/>.
- [3] *The Time-Travel Project*. <https://courses.softlab.ntua.gr/progds/2023b/time-travel/>.

