



# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

[www.cslab.ece.ntua.gr](http://www.cslab.ece.ntua.gr)

## ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ ΣΤΑ ΚΑΤΑΝΕΜΗΜΕΝΑ ΣΥΣΤΗΜΑΤΑ

Ακ. έτος 2023-2024, 9ο Εξάμηνο, Σχολή ΗΜ&ΜΥ

Τελική Ημερομηνία Παράδοσης: 3 εβδομάδες μετά το τέλος της εξεταστικής

### Εισαγωγή - Σκοπός

Το blockchain είναι η τεχνολογία πίσω από πολλές σύγχρονες αποκεντρωμένες εφαρμογές και αποτελεί στην πραγματικότητα μια κατανεμημένη βάση που επιτρέπει στους χρήστες της να κάνουν δοσοληψίες (transactions) μεταξύ τους με ασφάλεια, χωρίς να χρειάζονται κάποια κεντρική αρχή.

Σκοπός της εργασίας είναι να φτιάξετε το BlockChat, μια πλατφόρμα ανταλλαγής μηνυμάτων αλλά και καταγραφής δοσοληψιών που θα στηρίζεται σε ένα απλό blockchain, παρέχοντας στους χρήστες ασφαλή και αξιόπιστη αποστολή νομισμάτων είτε καταγραφή των μηνυμάτων που ανταλλάσσονται μεταξύ των συμμετεχόντων. Για το consensus θα εφαρμοστεί αλγόριθμος Proof-of-Stake.

Συνοπτικά, το σύστημά σας θα πρέπει να υλοποιεί τις παρακάτω λειτουργίες:

1. Ο κάθε χρήστης του BlockChat (το κάθε process δηλαδή) έχει ένα BlockChat wallet με coins, τα BCC (BlockChat Coins) που είναι απαραίτητα για να πραγματοποιεί transactions, δηλαδή αποστολές νομισμάτων είτε μηνυμάτων. Το κάθε wallet αποτελείται από: (α) ένα private key γνωστό μόνο στον χρήστη-κάτοχο του wallet, που του επιτρέπει να στέλνει νομίσματα/μηνύματα σε άλλους χρήστες (σε άλλα wallets δλδ), χρησιμοποιώντας BCC και (β) το αντίστοιχο public key, που απαιτείται για να λάβει ένας χρήστης (και κάτοχος wallet) νομίσματα/μήνυμα από άλλον χρήστη. Το public key είναι και η διεύθυνση του wallet του χρήστη.
2. Ο κάθε κάτοχος wallet μπορεί να κάνει συναλλαγές (νομίσματα/μηνύματα) ξοδεύοντας είδη των συναλλαγών είναι τα εξής:
  - a. Μεταφορά coins: Αν η Αλίκη θέλει να στείλει 9 BCC στον Μπόμπ, τότε θα δημιουργήσει ένα transaction που θα περιέχει το ποσό που θέλει να στείλει και τη διεύθυνση του wallet του Μπόμπ, δλδ το publickey του Μπόμπ. Το transaction αυτό θα το υπογράψει χρησιμοποιώντας το private key της. Η Αλίκη θα χρεωθεί ένα extra fee 3%.
  - b. Αποστολή μηνύματος: Αν η Αλίκη θέλει να στείλει ένα μήνυμα "Hello Bob" στον Μπόμπ, τότε θα δημιουργήσει ένα transaction που θα περιέχει (a) το String του μηνύματος που θέλει να στείλει, και (b) τη διεύθυνση του wallet του Μπόμπ, δλδ το public key του Μπόμπ. Το transaction αυτό θα το υπογράψει χρησιμοποιώντας το private key της. Η χρέωση για το

κάθε transaction είναι ανάλογη του μεγέθους του μηνύματος: 1BCC ανά χαρακτήρα μηνύματος. Οπότε η Αλίκη θα χρεωθεί 9BCC.

3. Το κάθε transaction που δημιουργείται γίνεται broadcast σε όλο το δίκτυο BlockChat.
4. Οι κόμβοι που συμμετέχουν στη διαδικασία της επικύρωσης blocks, μόλις λάβουν το νέο transaction το επικυρώνουν: Στο παράδειγμά μας, πρώτα ελέγχουν ότι το transaction προήλθε από την Αλίκη χρησιμοποιώντας το public key της, πιστοποιούν ότι η Αλίκη έχει αρκετά χρήματα στο wallet της για να εκτελέσει το transaction (9BCC στην περίπτωση μας) και αν όλοι οι έλεγχοι επιτύχουν, προσθέτουν το transaction στο τρέχον block. Για την παρούσα άσκηση θεωρούμε ότι όλοι οι χρήστες συμμετέχουν στη διαδικασία επικύρωσης.
5. Όταν το τρέχον block γεμίσει, τότε οι κόμβοι ξεκινούν τη διαδικασία της επικύρωσης ακολουθώντας αλγόριθμο Proof-of-Stake που περιγράφεται παρακάτω. Όποιος τελικά επικυρώσει το block, το κάνει broadcast σε όλο το δίκτυο, ώστε να γίνει γνωστό σε όλους τους κόμβους που συμμετέχουν.
6. Ο επικυρωτής του block είναι και αυτός στου οποίου το wallet πιστώνονται οι χρεώσεις των transactions που έχουν συμπεριληφθεί στο block .

## Δίκτυο

Υποθέτουμε ότι το δίκτυό μας έχει μέγεθος  $n$ , δηλαδή  $n$  κόμβοι συμμετέχουν στο δίκτυο με ids από 0 έως  $n-1$ . Ο κάθε κόμβος του δικτύου μας θα πρέπει να μπορεί να πραγματοποιεί συναλλαγές με όλους τους υπόλοιπους κόμβους του συστήματος γνωρίζοντας την ip address και το port στο οποίο ακούνε. Για λόγους απλότητας, θεωρούμε ότι κάθε κόμβος διατηρεί λίστα με όλους τους υπόλοιπους κόμβους που συμμετέχουν κάθε στιγμή στο BlockChat σύστημα. Η επικοινωνία μπορεί να υλοποιηθεί είτε με sockets (TCP ή UDP) είτε με REST API, χρησιμοποιώντας όποια βιβλιοθήκη θέλετε.

Ο πρώτος κόμβος του δικτύου λέγεται bootstrap node (id 0) και είναι ο κόμβος που δημιουργεί το genesis block, το πρώτο δηλ block του blockchain μας. Στο genesis block θέτουμε previous\_hash=1, validator=0 και η λίστα από transactions περιλαμβάνει μόνο ένα transaction που δίνει στον bootstrap κόμβο  $1000 \cdot n$  BCC coins από την wallet διεύθυνση 0. Αυτό είναι το μοναδικό block που δεν επαληθεύεται. Σταδιακά προστίθενται οι υπόλοιποι  $n-1$  κόμβοι του δικτύου.

Για να εισαχθεί ένας νέος κόμβος στο σύστημα επικοινωνεί πρώτα με τον bootstrap κόμβο, του οποίου θεωρούμε γνωστά σε όλους τα στοιχεία επικοινωνίας ip address/port, και του στέλνει το public key του wallet του. Από τον bootstrap κόμβο λαμβάνει το μοναδικό id του (ο πρώτος το id 1, ο δεύτερος το id 2 κ.ο.κ) . Όταν εισαχθούν όλοι οι κόμβοι, ο bootstrap κάνει broadcast σε όλους τα ζεύγη ip address/port καθώς και τα public keys των wallets όλων των κόμβων που συμμετέχουν στο σύστημα. Θεωρείστε ότι από εδώ και πέρα δεν υπάρχουν εισαγωγές ή αποχωρήσεις κόμβων από το σύστημα. Επιπλέον, κάθε νέος κόμβος λαμβάνει από τον bootstrap κόμβο το blockchain όπως έχει διαμορφωθεί μέχρι εκείνη τη στιγμή, το κάνει validate και μετά από αυτό το σημείο μπορεί να κάνει transactions.

Ο bootstrap κόμβος, όπως αναφέρθηκε παραπάνω, έχει  $1000 \cdot n$  BCC coins από το genesis block. Μετά τη σύνδεση κάθε νέου κόμβου στο δίκτυο, ο bootstrap κόμβος εκτελεί ένα transaction όπου

του μεταφέρει 1000 BCC. Έτσι, μετά την εισαγωγή όλων των κόμβων, ο καθένας τους έχει 1000 BCC στο wallet του.

## ChatBlock backend

Στο ChatBlock backend υλοποιείται όλη η λογική του blockchain. Τα βασικά συστατικά στοιχεία του blockchain είναι τα παρακάτω:

### Block

Το κάθε block έχει τις εξής πληροφορίες

- index: ο αύξων αριθμός του block,
- timestamp: το timestamp της δημιουργίας του block
- transactions: Η λίστα με τα transactions που περιέχονται στο block
- validator: το public key του κόμβου που επικύρωσε το block
- current\_hash: το hash του block
- previous\_hash: το hash του προηγούμενου block στο blockchain.

Θεωρούμε ότι το κάθε block έχει συγκεκριμένη χωρητικότητα σε αριθμό από transaction. Η χωρητικότητα καθορίζεται από τη σταθερά capacity.

### Blockchain

Η λίστα από blocks που έχουν επαληθευτεί.

### Wallet

Ένα wallet συνδέεται με ένα ζεύγος public/private key. Το public key δρα ως διεύθυνση του wallet, την οποία μπορεί κανείς να μοιραστεί με οποιονδήποτε προκειμένου να μπορεί να δεχτεί μηνύματα. Το private key χρησιμοποιείται για να υπογράφονται τα transactions, ώστε να εξασφαλίζεται ότι μόνο ο κάτοχος του wallet (δλδ ο κάτοχος του αντίστοιχου private key) μπορεί να ξοδέψει χρήματα από το wallet στέλνοντας κάποιο μήνυμα. Αυτό γίνεται με τη συνάρτηση `sign_transaction()`. Οποιοσδήποτε γνωρίζει το public key ενός wallet μπορεί να επαληθεύσει ότι ένα transaction που με αποστολέα το wallet αυτό έχει δημιουργηθεί από τον κάτοχό του με την `verify_transaction()`.

### Transaction

Κάθε transaction περιέχει πληροφορίες για την αποστολή νομισμάτων/μηνύματος από ένα wallet σε ένα άλλο. Οι πληροφορίες που περιλαμβάνει είναι

sender\_address: Το public key του wallet από το οποίο προέρχεται το μήνυμα

receiver\_address: Το public key του wallet παραλήπτη του μηνύματος

type\_of\_transaction: Καθορίζει τον τύπο του transaction (coins or message)

amount: το ποσο νομισμάτων προς αποστολή

message: το String του μηνύματος που στέλνεται

nonce: counter που διατηρείται ανά αποστολέα και αυξάνεται κατά 1 μετά από κάθε αποστολή

transaction\_id: το hash του transaction

Signature: Υπογραφή που αποδεικνύει ότι ο κάτοχος του wallet δημιούργησε αυτό το transaction.

Ένα transaction μπορεί να δημιουργηθεί από τον κάτοχο του wallet, ο οποίος θα χρεωθεί είτε το amount που θέλει να στείλει είτε το κόστος του μηνύματος. Κάθε transaction γίνεται broadcast σε όλα τα μέλη του blockchain. Κατά τη λήψη ενός transaction από οποιονδήποτε node, καλείται η συνάρτηση `validate_transaction` που ελέγχει την ορθότητά του. Κάθε transaction στο blockchain στα πλαίσια της ασκήσης θα πρέπει να ανήκει σε 1 από τα δύο use cases (είτε θα στέλνει coins είτε message). Ωστόσο αν κάποιος επιθυμεί να υλοποιήσει transaction που να μπορεί να στείλει και μήνυμα και νομίσματα μπορεί να το υλοποιήσει. Επιπλέον αν σας βολεύει περισσότερο μπορείτε να ορίσετε δύο (αντί για έναν όπως φαίνεται παραπάνω) διαφορετικούς τύπους transactions στον κώδικα.

### **Proof-of-Stake**

Το Proof of Stake (PoS) είναι ένας μηχανισμός consensus, ο οποίος αναπτύσσει μια εναλλακτική προσέγγιση σε σύγκριση με τον πιο γνωστό Proof of Work (PoW). Στο PoS, το ποιος θα επικυρώσει το επόμενο μπλοκ δεν εξαρτάται από την υπολογιστική δύναμη, αλλά από το ποσό κρυπτονομίσματος (staking) που κατέχει ένας κόμβος. Στην πράξη, κάθε κόμβος που θέλει να συμμετάσχει στη διαδικασία του validation, δεσμεύει για τον σκοπό αυτό το ποσό που θέλει από το πορτοφόλι του με τη συνάρτηση `stake(amount)`. Αυτή η κλήση μεταφράζεται σε ένα transaction του οποίου το `receiver_address` είναι 0 και το ποσό που θέλει να δεσμεύσει ο κάθε κόμβος. Κατά τη λήψη τέτοιων transactions θα πρέπει να ελέγχεται εκτός των άλλων ότι υπάρχει το staking ποσό στο wallet του αποστολέα.

Η πιθανότητα να είναι ο επόμενος validator είναι ανάλογη του ποσού που δέσμευσε σε σχέση με το συνολικό ποσό που έχει δεσμευτεί από όλους τους πιθανούς validators. Αν επομένως ένας κόμβος έχει το 5% του συνολικού staking ποσού, τότε έχει και 5% πιθανότητα να επιλεγεί ως validator για το επόμενο block. Η διαδικασία είναι η εξής: Μόλις συμπληρωθεί το τρέχον block με capacity transactions, τότε ο κόμβος καλεί μια ψευδοτυχαία γεννήτρια αριθμών που θα του υποδείξει ποιος θα είναι ο validator του block. Για να είναι ντετερμινιστικό το αποτέλεσμα της κλήσης της από οποιονδήποτε κόμβο, πρέπει να χρησιμοποιηθεί από όλους τους κόμβους το ίδιο seed, το οποίο καθορίζεται από το hash του προηγούμενου block της αλυσίδας. Ο κόμβος που αναγνωρίζει τον εαυτό του ως validator επικυρώνει το block και το στέλνει με broadcast σε όλους.

Για την εύρεση του validator από την ψευδοτυχαία γεννήτρια με βάση τις πιθανότητες του κάθε κόμβου να επιλεγεί, μπορεί να ακολουθηθεί η λογική της «λοτταρίας»: Ο κάθε κόμβος βάζει στον κουβά τόσους λαχνούς, όσα BCC έχει δεσμεύσει, δίνοντάς τους διαδοχικούς αριθμούς. Η γεννήτρια

θα τραβήξει έναν λαχνό, δηλ ένα νούμερο από το 1 μέχρι τον συνολικό αριθμό των δεσμευμένων BCC και έτσι θα καθορίσει τον επόμενο validator.

### **Μοντέλο δεδομένων**

Θα ακολουθηθεί το μοντέλο λογαριασμών (account model) για τη διατήρηση των δεδομένων κατάστασης (state) του blockchain. Το state επομένως που πρέπει να διατηρεί κάθε κόμβος του δικτύου είναι η λίστα με τους υπάρχοντες λογαριασμούς και τα υπόλοιπά τους. Επίσης θα πρέπει να διατηρείται πληροφορία με το staking του κάθε κόμβου (το οποίο πρέπει να αφαιρείται από το διαθέσιμο υπόλοιπο του λογαριασμού). Με κάθε συναλλαγή, οι κόμβοι που τη λαμβάνουν πρέπει να μειώσουν το υπόλοιπο του λογαριασμού του αποστολέα σύμφωνα με το ποσό των νομισμάτων που επιθυμεί να στείλει ή με τον κανόνα χρέωσης των μηνυμάτων (1BCC για κάθε χαρακτήρα). Μετά τη λήψη ενός επικυρωμένου block, ο κάθε κόμβος πρέπει να αυξήσει το υπόλοιπο του validator:

1. Τα συνολικά fees που χρεώθηκαν οι αποστολείς transactions που αφορούν σε μεταφορά χρημάτων και περιλαμβάνονται στο block.
2. Το σύνολο των BCC που χρεώθηκαν οι αποστολείς όλων των μηνυμάτων που εμπεριέχονται στο block.

Για την πρόληψη επιθέσεων επανάληψης (replay attacks), όπου κάποιος κακόβουλος κόμβος θα μπορούσε να ξαναστείλει ένα transaction που έλαβε για να χρεώσει ξανά τον αποστολέα, κάθε συναλλαγή στο μοντέλο λογαριασμού περιέχει ένα πεδίο nonce. Το nonce είναι ένας μετρητής που διατηρεί κάθε λογαριασμός στο BlockChat και που αυξάνεται κατά ένα με κάθε εξερχόμενη συναλλαγή. Αυτό εμποδίζει την υποβολή της ίδιας συναλλαγής περισσότερες από μία φορές στο δίκτυο.

### **Βασικές συναρτήσεις**

Οι βασικές λειτουργίες του συστήματος περιλαμβάνουν:

`generate_wallet()`

Δημιουργεί νέο wallet, δηλ ζεύγος public/private key χρησιμοποιώντας τον κρυπτογραφικό αλγόριθμο RSA.

`create_transaction()`

Δημιουργείται ένα νέο transaction που περιέχει όλα τα στοιχεία που απαιτούνται.

`sign_transaction()`

Υπογράφεται το transaction με το private key του wallet.

`broadcast_transaction()`

Το transaction αποστέλλεται με broadcast σε όλους τους κόμβους.

`verify_signature()`

Επαληθεύεται η υπογραφή του transaction αμέσως μετά τη λήψη του.

`validate_transaction()`

Επαληθεύεται η ορθότητα του transaction που έχει ληφθεί. Η επαλήθευση περιλαμβάνει (α) την επαλήθευση της υπογραφής (`verify_signature`) και (β) τον έλεγχο του υπολοίπου του λογαριασμού του αποστολέα για να εξασφαλίσουμε ότι έχει το ποσό που απαιτείται για να πραγματοποιήσει το transaction (είτε τα coins που καθορίζονται από το πεδίο `amount`, είτε τα coins που αντιστοιχούν στους χαρακτήρες του μηνύματος που θέλει να στείλει). Θα πρέπει να ληφθεί υπόψιν και το ποσό που έχει δεσμευτεί για staking (αυτό δεν πρέπει να ξοδευτεί).

`mint_block()`

Η συνάρτηση αυτή καλείται μόλις capacity transactions έχουν ληφθεί και επαληθευτεί από κάποιον κόμβο και υλοποιεί το proof of stake καλώντας την ψευδοτυχαία γεννήτρια αριθμών με το κατάλληλο seed. Αν ο κόμβος που την κάλεσε είναι ο validator, γεμίζει όλα τα πεδία του block με τις κατάλληλες πληροφορίες.

`broadcast_block()`

Μόλις το block δημιουργηθεί από τον validator, τότε αυτός κάνει broadcast το επαληθευμένο block σε όλους τους υπόλοιπους κόμβους.

`validate_block()`

Αυτή η συνάρτηση καλείται από τους nodes κατά τη λήψη ενός νέου block (εκτός του genesis block). Επαληθεύεται ότι (a) ο validator είναι πράγματι ο σωστός (αυτός που υπέδειξε η κλήση της ψευδοτυχαίας γεννήτριας) και ότι (b) το πεδίο `previous_hash` ισούται πράγματι με το hash του προηγούμενου block.

`validate_chain()`

Αυτή η συνάρτηση καλείται από τους νεοεισερχόμενους κόμβους, οι οποίοι επαληθεύουν την ορθότητα του blockchain που λαμβάνουν από τον bootstrap κόμβο. Στην πραγματικότητα καλείται η `validate_block()` για όλα τα blocks εκτός του genesis.

`stake(amount)`

Η συνάρτηση καλείται από τους nodes για να καθορίσουν το ποσό που δεσμεύουν ως stake για τη διαδικασία του proof-of-stake. Αυτό το ποσό καθορίζει και την πιθανότητα του κόμβου να επιλεγεί ως validator. Εννοείται ότι κάθε validator θα πρέπει να είναι σε θέση να κάνει και update στο ποσό που έχει αποφασίσει να δεσμεύσει.

## BlockChat client

Θα πρέπει να υλοποιήσετε έναν client (ένα απλό cli αρκεί, αλλά αν θέλετε να είστε πιο fancy θα μετρήσει θετικά) που θα δίνει στον χρήστη τη δυνατότητα να εκτελεί τα παρακάτω:

*t <recipient\_address> <message>*

New transaction: Στείλε στο `recipient_address` wallet το ποσό `amount` από BTC coins που θα πάρει από το `wallet sender_address`. Θα καλεί συνάρτηση `create_transaction` στο backend που θα υλοποιεί την παραπάνω λειτουργία.

*t <recipient\_address> <message>*

New message: Στείλε στο recipient\_address wallet το μήνυμα message χρεώνοντας κατάλληλα το wallet του αποστολέα. Θα καλεί συνάρτηση create\_transaction στο backend που θα υλοποιεί την παραπάνω λειτουργία.

*stake <amount>*

Set the node stake: Δέσμευσε amount ποσό για staking του συγκεκριμένου κόμβου. Καλεί την stake(amount) όπως ορίστηκε παραπάνω.

*view*

View last block: Τύπωσε τα transactions που περιέχονται στο τελευταίο επικυρωμένο block του BlockChat blockchain καθώς και το id του validator του block αυτού. Καλεί τη συνάρτηση view\_block() στο backend που υλοποιεί την παραπάνω λειτουργία.

*balance*

Show balance: Τύπωσε το υπόλοιπο του wallet.

*help*

Επεξήγηση των παραπάνω εντολών.

## Πειράματα

Θα αναπτύξετε το BlockChat σε όποια γλώσσα προγραμματισμού θέλετε. Θα το στήσετε σε υποδομή του ~okeanos (θα σας δοθούν ακριβείς οδηγίες). Για την αναφορά θα εκτελέσετε τα παρακάτω πειράματα:

### 1) Απόδοση του συστήματος

Θα στήσετε ένα BlockChat με 5 clients. Αφού όλοι εισέλθουν στο σύστημα, ο καθένας θα διαβάσει το αρχείο transX.txt, όπου X το node id του κόμβου. Το κάθε αρχείο περιέχει μηνύματα προς άλλους κόμβους με τη μορφή

<recipient\_node\_id> <message string>

Έτσι ο κάθε κόμβος δημιουργεί και στέλνει στο δίκτυο ένα transaction ανά γραμμή. Η διαδικασία αυτή θα γίνει ταυτόχρονα για όλους τους clients. Με σταθερό staking 10 BCC για κάθε κόμβο και capacity 5, 10 και 20 θα καταγράψετε τα παρακάτω:

- Throughput (ρυθμαπόδοση) του συστήματός σας, δηλαδή πόσα transactions εξυπηρετούνται στην μονάδα του χρόνου.

- Block time, δηλ τον μέσο χρόνο που απαιτείται για να προστεθεί ένα νέο block στο blockchain.

## 2) Κλιμακωσιμότητα του συστήματος

Θα επαναλάβετε το πείραμα για 10 clients και θα συγκρίνετε με τα προηγούμενα αποτελέσματα παρουσιάζοντας σε γράφημα τις μετρικές του προηγούμενου πειράματος (άξονας y) σε σχέση με τον αριθμό των clients (άξονας x).

## 3) Δικαιοσύνη

Επαναλάβετε το πείραμα με τους 5 κόμβους και capacity 5, μόνο που αυτή τη φορά ένα από όλους τους κόμβους έχει staking 100 BCC. Τι παρατηρείτε σε σχέση με τη δικαιοσύνη του συστήματος;

Δουλέψτε σε ομάδες 2-3 ατόμων. Παραδοτέο της άσκησης θα είναι ο πηγαίος κώδικας (tarball με τα σχετικά αρχεία) καθώς και ένα ηλεκτρονικό κείμενο (pdf, docx ή odt) που θα παρουσιάζει τον σχεδιασμό του συστήματός σας και τα αποτελέσματα των πειραμάτων. Ο κώδικας και η αναφορά θα παραδοθούν ηλεκτρονικά στην ιστοσελίδα του μαθήματος. Επίδειξη της άσκησης θα γίνει σε συνεννόηση με τους διδάσκοντες μετά τη λήξη της προθεσμίας για το παραδοτέο.