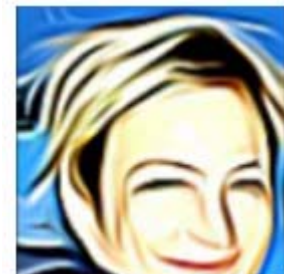


Απόκτηση και καλλιέργεια Υπολογιστικής Σκέψης

- Διακριτά Μαθηματικά
- Εισαγωγή στους Αλγόριθμους
- Αλγοριθμικά Θέματα Ασύρματων Δικτύων
- Υπολογιστική Επιστήμη και Πολιτισμός

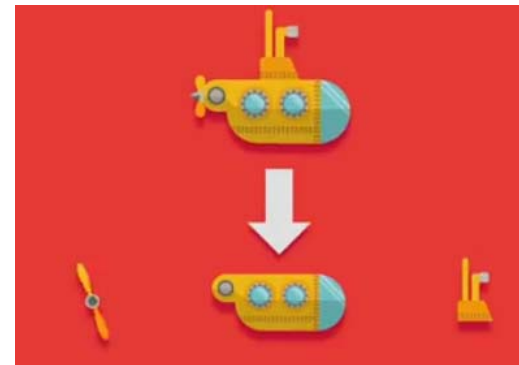


Τι είναι η υπολογιστική σκέψη;

- Οι υπολογιστές μας βοηθάνε στην υλοποίηση λύσεων για προβλήματα αλλά...
- Πριν να έχουμε τη λύση ενός προβλήματος πρέπει να κατανοήσουμε το ίδιο το πρόβλημα και τους τρόπους επίλυσής του
- Η απόκτηση και καλλιέργεια υπολογιστικής σκέψης μας βοηθάει στο να:
 - να κατανοούμε περίπλοκα προβλήματα,
 - να επινοούμε λύσεις για αυτά και
 - να μπορούμε να παρουσιάσουμε τις λύσεις αυτές με τρόπο κατανοητό από ανθρώπους και υπολογιστές

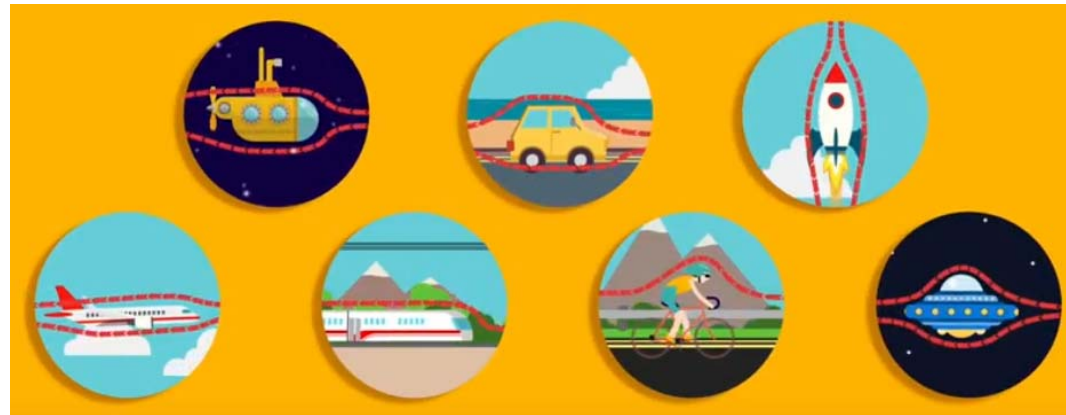
Οι 4 ακρογωνιαίοι λίθοι της υπολογιστικής σκέψης

- Υπάρχουν 4 βασικές τεχνικές στις οποίες βασίζεται η υπολογιστική σκέψη:
 - **Αποσύνθεση (decomposition)**: η κατάτμηση ενός περίπλοκου προβλήματος ή συστήματος σε άλλα μικρότερα μέρη ευκολότερα διαχειρίσιμα



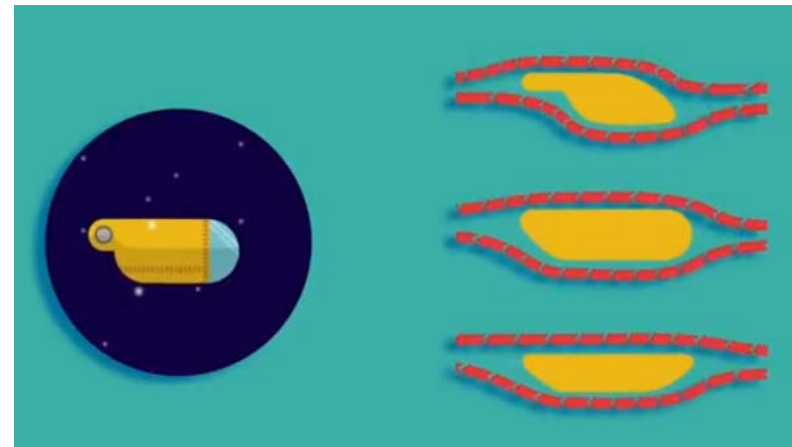
Οι 4 ακρογωνιαίοι λίθοι της υπολογιστικής σκέψης

- Υπάρχουν 4 βασικές τεχνικές στις οποίες βασίζεται η υπολογιστική σκέψη:
 - **Αποσύνθεση (decomposition)**: η κατάτμηση ενός περίπλοκου προβλήματος ή συστήματος σε άλλα μικρότερα μέρη ευκολότερα διαχειρίσιμα
 - **Αναγνώριση προτύπων (pattern recognition)**: εντοπισμός ομοιοτήτων μεταξύ τέτοιων προβλημάτων ή υπο-προβλημάτων



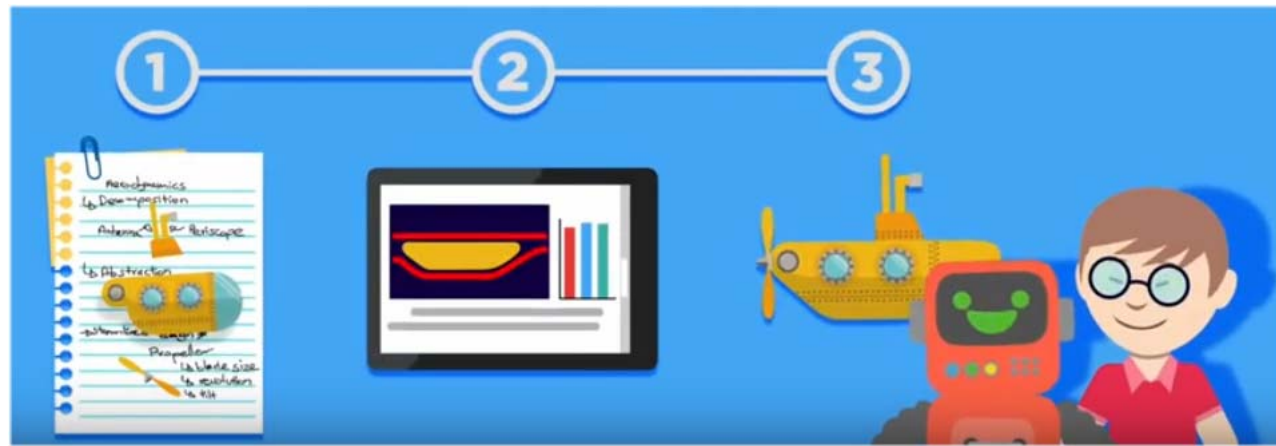
Οι 4 ακρογωνιαίοι λίθοι της υπολογιστικής σκέψης

- Υπάρχουν 4 βασικές τεχνικές στις οποίες βασίζεται η υπολογιστική σκέψη:
 - **Αποσύνθεση (decomposition)**: η κατάτμηση ενός περίπλοκου προβλήματος ή συστήματος σε άλλα μικρότερα μέρη ευκολότερα διαχειρίσιμα
 - **Αναγνώριση προτύπων (pattern recognition)**: εντοπισμός ομοιοτήτων μεταξύ τέτοιων προβλημάτων ή υπο-προβλημάτων
 - **Αφαίρεση (abstraction)**: προσήλωση στην ουσιώδη πληροφορία αγνοώντας περιττές λεπτομέρειες



Οι 4 ακρογωνιαίοι λίθοι της υπολογιστικής σκέψης

- Υπάρχουν 4 βασικές τεχνικές στις οποίες βασίζεται η υπολογιστική σκέψη:
 - **Αποσύνθεση (decomposition)**: η κατάτμηση ενός περίπλοκου προβλήματος ή συστήματος σε άλλα μικρότερα μέρη ευκολότερα διαχειρίσιμα
 - **Αναγνώριση προτύπων (pattern recognition)**: εντοπισμός ομοιοτήτων μεταξύ τέτοιων προβλημάτων ή υπο-προβλημάτων
 - **Αφαίρεση (abstraction)**: προσήλωση στην ουσιώδη πληροφορία αγνοώντας περιττές λεπτομέρειες
 - **Αλγόριθμοι (algorithms)**: ανάπτυξη λύσεων βήμα προς βήμα ή κανόνων για την επίλυση προβλημάτων



Οι 4 ακρογωνιαίοι λίθοι της υπολογιστικής σκέψης

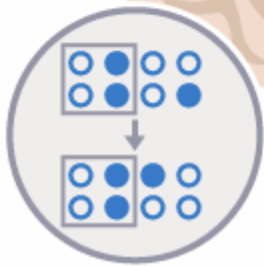
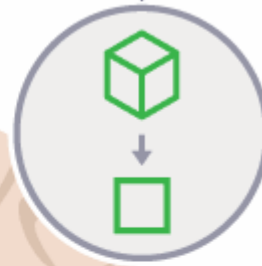
- Υπάρχουν 4 βασικές τεχνικές στις οποίες βασίζεται η υπολογιστική σκέψη:
 - **Αποσύνθεση (decomposition)**: η κατάτμηση ενός περίπλοκου προβλήματος ή συστήματος σε άλλα μικρότερα μέρη ευκολότερα διαχειρίσιμα
 - **Αναγνώριση προτύπων (pattern recognition)**: εντοπισμός ομοιοτήτων μεταξύ τέτοιων προβλημάτων ή υπο-προβλημάτων
 - **Αφαίρεση (abstraction)**: προσήλωση στην ουσιώδη πληροφορία αγνοώντας περιττές λεπτομέρειες
 - **Αλγόριθμοι (algorithms)**: ανάπτυξη λύσεων βήμα προς βήμα ή κανόνων για την επίλυση προβλημάτων
- Κάθε μία από τις 4 τεχνικές είναι εξίσου σημαντική – σαν πόδια ενός τραπεζιού – αν κάποιος λείπει το τραπέζι δεν μπορεί να σταθεί – η σωστή εφαρμογή αυτών των 4 τεχνικών βοηθάει στη σωστή καθοδήγηση ενός “υπολογιστή” – ατόμου ή μηχανής (προγραμματισμό) – για την υλοποίηση της λύσης του προβλήματος

Computational thinking

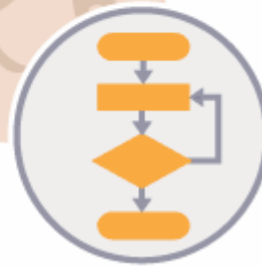
Decomposition



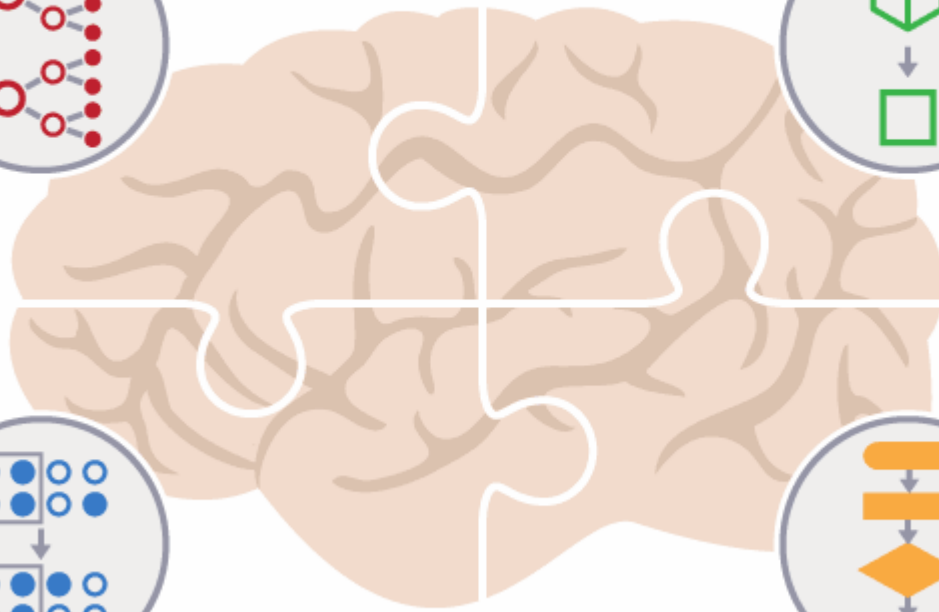
Abstraction



Pattern recognition



Algorithms

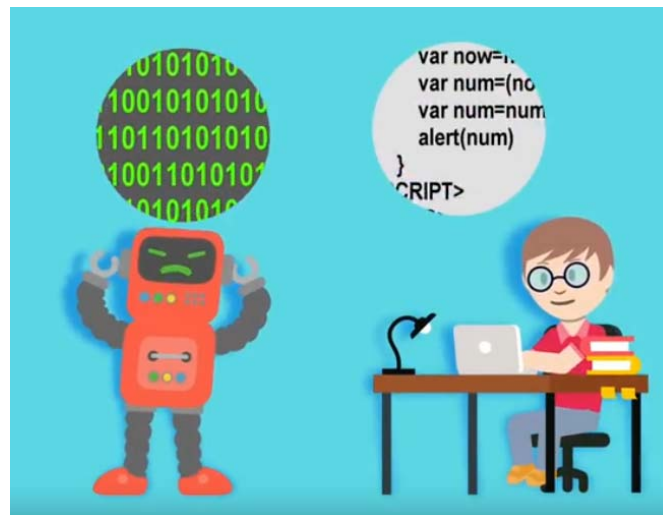


Η υπολογιστική σκέψη στην πράξη

- Ένα περίπλοκο πρόβλημα έχει το χαρακτηριστικό ότι εκ πρώτης όψεως δεν έχουμε ιδέα πώς να το λύσουμε
- Η υπολογιστική σκέψη συντελεί στο να μάθουμε να σκεφτόμαστε ως εξής:
 - Αρχικά, «σπάμε» το δύσκολο πρόβλημα σε άλλα μικρότερα, ευκολότερα διαχειρίσιμα προβλήματα (**decomposition**)
 - Ασχολούμαστε ξεχωριστά με καθένα από αυτά τα λιγότερο περίπλοκα προβλήματα εξετάζοντας πώς έχουμε λύσει ήδη άλλα παραπλήσια προβλήματα (**pattern recognition**)
 - Εστιάζουμε μόνο στα ουσιώδη χαρακτηριστικά αγνοώντας τις λεπτομέρειες (**abstraction**)
 - Μετά, είμαστε σε θέση να επινοήσουμε απλά βήματα ή κανόνες για να λύσουμε τα λιγότερο περίπλοκα υποπροβλήματα (**algorithms**)
- Αυτά τα απλά βήματα ή κανόνες χρησιμοποιούνται και για την ανάπτυξη προγραμμάτων (λογισμικού) μέσω των οποίων θα αξιοποιήσουμε τους υπολογιστές για την επίλυση πολύπλοκων προβλημάτων

Απόκτηση και καλλιέργεια υπολογιστικής σκέψης

- Η υιοθέτηση υπολογιστικής σκέψης δεν είναι προγραμματισμός
- Δεν είναι ούτε καν σαν να σκέπτεται κανείς σαν υπολογιστής αφού οι υπολογιστές δεν διαθέτουν (ακόμα) πρωτοτυπία, δηλ., δεν σκέπτονται – δεν μπορούν να σκεφτούν

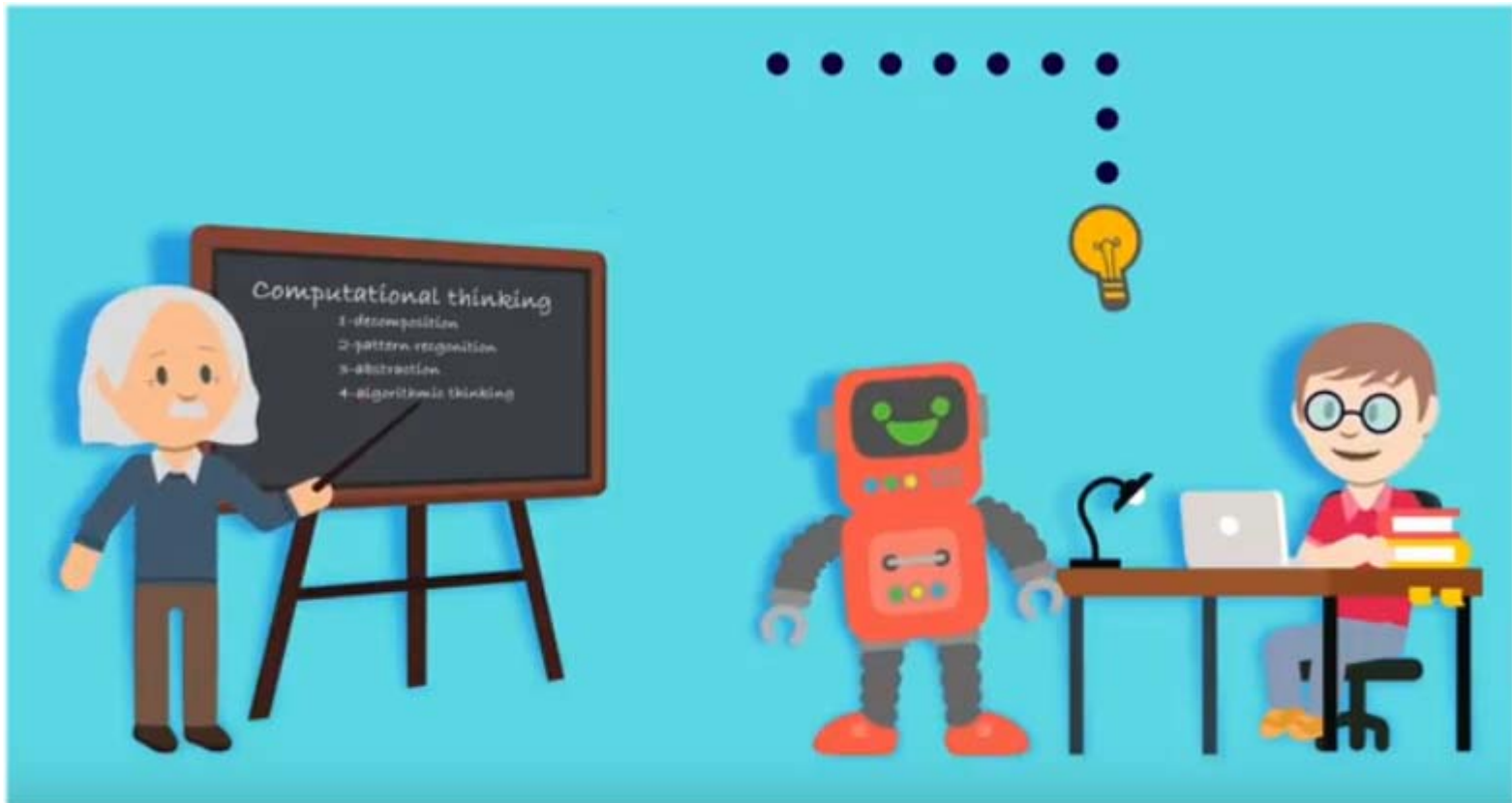


Απόκτηση και καλλιέργεια υπολογιστικής σκέψης

- Η υιοθέτηση υπολογιστικής σκέψης δεν είναι προγραμματισμός
- Δεν είναι ούτε καν σαν να σκέπτεται κανείς σαν υπολογιστής αφού οι υπολογιστές δεν διαθέτουν (ακόμα) πρωτοτυπία, δηλ., δεν σκέπτονται – δεν μπορούν να σκεφτούν
- Με απλά λόγια, μέσω του προγραμματισμού λέμε σε έναν υπολογιστή τι και πώς να το κάνει
- **Η υπολογιστική σκέψη δίνει τη δυνατότητα να επινοήσουμε ακριβώς τι να πούμε σε έναν υπολογιστή να κάνει**

Απόκτηση και καλλιέργεια υπολογιστικής σκέψης

Κριτική και λογική σκέψη για τη συστηματική επίλυση προβλημάτων

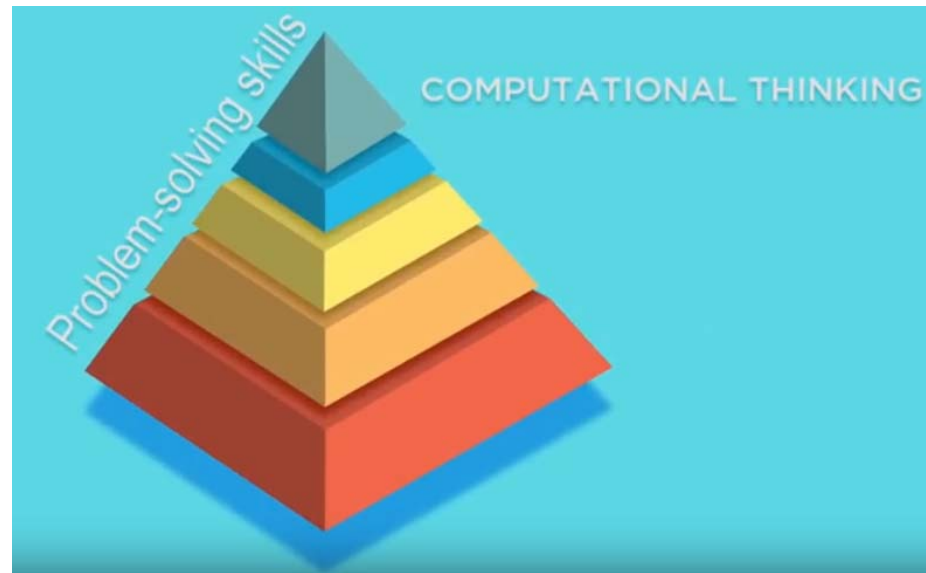


Υπολογιστική σκέψη δεν είναι μόνον προγραμματισμός

- Φανταστείτε ότι συμφωνήσατε να συναντηθείτε με τους φίλους σας ένα μέρος που δεν έχετε πάει ποτέ ξανά και πριν φύγετε από το σπίτι σχεδιάζετε τη διαδρομή που θα ακολουθήσετε
- Ίσως σκέφτεστε τις διαθέσιμες διαδρομές και ποια είναι η προτιμότερη («βέλτιστη») – αυτή μπορεί να είναι
 - η συντομότερη από πλευράς απόστασης διαδρομή
 - η λιγότερο χρονοβόρα διαδρομή
 - η διαδρομή που περνάει από το αγαπημένο σας coffee shop
- Θα ακολουθήσετε τότε οδηγίες βήμα-βήμα για να φτάσετε στον προορισμό σας μέσω της διαδρομής που επιλέξατε
- Στην περίπτωση αυτή:
 - το κομμάτι του **σχεδιασμού της διαδρομής** αποτελεί αναλογία για την **υπολογιστική σκέψη**
 - το να ακολουθήσετε τις **οδηγίες για τη μετάβαση** αποτελεί αναλογία για τον **προγραμματισμό**

Υπολογιστική σκέψη και καθημερινότητα

- Η ικανότητα μετατροπής ενός πολύπλοκου προβλήματος σε ένα εύκολα κατανοητό πρόβλημα είναι μια εξαιρετικά χρήσιμη δεξιότητα
- Στα αλήθεια, είναι μια δεξιότητα που ήδη έχετε και χρησιμοποιείτε καθημερινά



Υπολογιστική σκέψη και καθημερινότητα: συνάντηση με φίλους

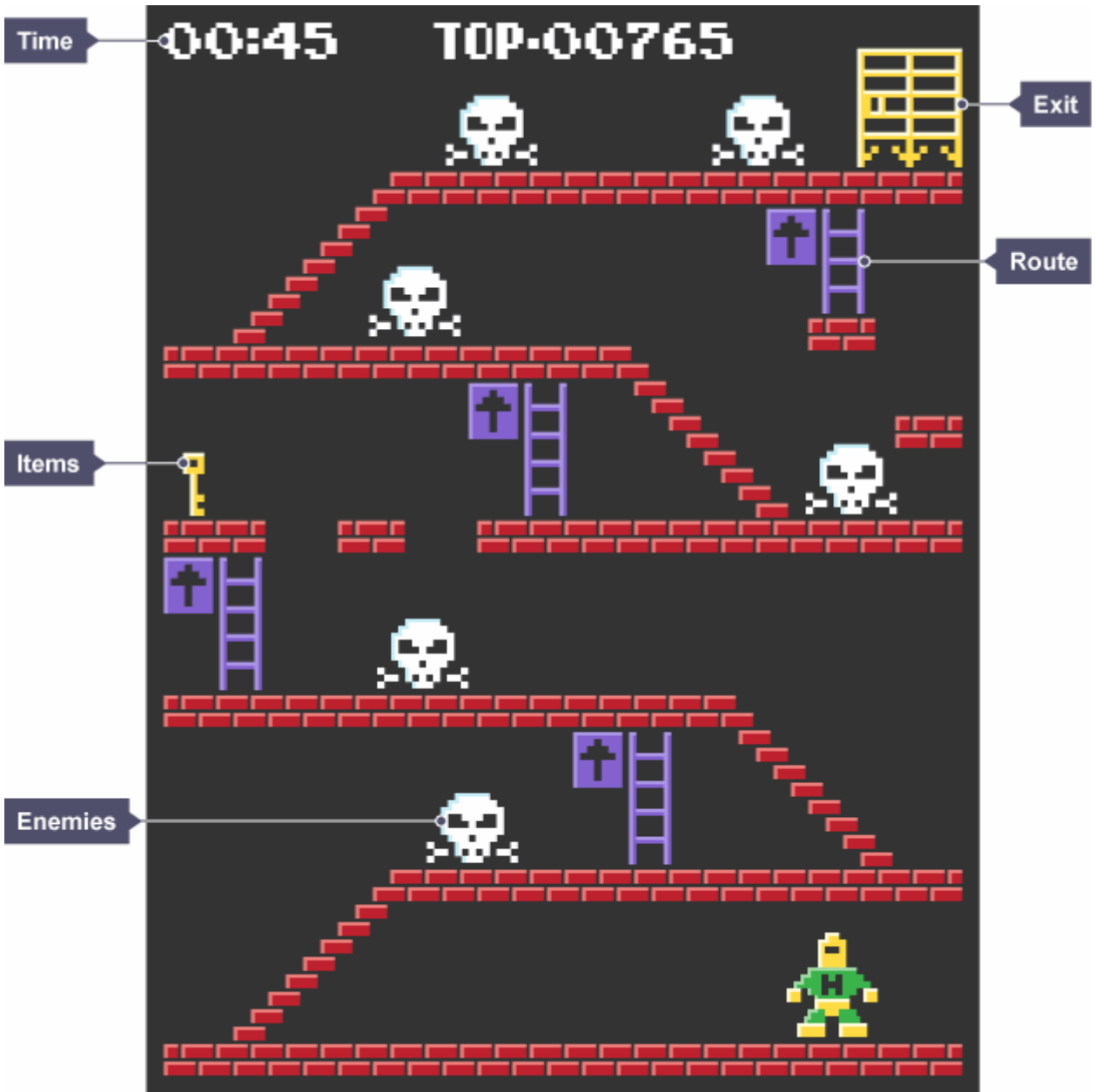
- Φανταστείτε ότι θέλετε να αποφασίσετε τι θα κάνετε με την παρέα σας
- Αν όλοι έχετε διαφορετικές προτιμήσεις, θα πρέπει να αποφασίσετε:
 - Τι θα μπορούσατε να κάνετε
 - Πού θα μπορούσατε να πάτε
 - Ποιος θέλει να κάνει τι
 - Τι έχετε κάνει στο παρελθόν που είχε επιτυχή έκβαση
 - Πόσα χρήματα διαθέτετε και ποιο είναι το κόστος της κάθε εναλλακτικής πρότασης
 - Πώς θα είναι ο καιρός
 - Πόσο χρόνο έχετε στη διάθεσή σας
 - ...

Υπολογιστική σκέψη και καθημερινότητα: συνάντηση με φίλους

- Από αυτές τις πληροφορίες, θα μπορούσατε ευκολότερα να αποφασίσετε πού να πάτε και τι να κάνετε – ώστε τα περισσότερα άτομα να είναι χαρούμενα
- Θα μπορούσατε επίσης – αν θέλετε - να χρησιμοποιήσετε υπολογιστή για να συλλέξετε και να αναλύσετε τα δεδομένα ώστε να καταλήξετε στη βέλτιστη λύση για το πρόβλημα, τόσο στη συγκεκριμένη περίπτωση όσο και σε άλλες παραπλήσιες που ενδεχομένως ανακύψουν στο μέλλον

Υπολογιστική σκέψη και καθημερινότητα: ηλεκτρονικό παιχνίδι

- Φανταστείτε ότι παίζετε ένα ηλεκτρονικό παιχνίδι
- Ανάλογα με το παιχνίδι, για να περάσετε την πίστα πρέπει να γνωρίζετε:
 - Τι αντικείμενα πρέπει να συλλέξετε, πώς να τα συλλέξετε και πόσο χρόνο έχετε στη διάθεσή σας για τα συλλέξετε
 - Πού είναι η έξοδος και ποια είναι η καλύτερη διαδρομή ώστε να φτάσετε γρηγορότερα στην έξοδο
 - Τι είδους αντίπαλοι υπάρχουν και τα αδύνατα σημεία τους
- Από αυτές τις λεπτομέρειες μπορείτε να επινοήσετε μια στρατηγική για να περάσετε την πίστα με τον πιο αποδοτικό τρόπο



Υπολογιστική σκέψη και καθημερινότητα

- Και τα δύο προηγούμενα παραδείγματα είναι ενδεικτικά για τις περιπτώσεις στις οποίες χρησιμοποιείται η υπολογιστική σκέψη για την επίλυση πολύπλοκων προβλημάτων:
 - Κάθε πολύπλοκο πρόβλημα αποσυντίθεται σε απλούστερες αποφάσεις και βήματα (π.χ., πού να πάτε, πώς να περάσετε την πίστα – decomposition)
 - Δίνεται έμφαση μόνο στις σημαντικές λεπτομέρειες (π.χ., καιρός, τοποθεσία ή έξοδος – abstraction)
 - Αξιοποιείται γνώση για παραπλήσια προβλήματα που έχουν ήδη αντιμετωπιστεί (pattern recognition) ...
 - ... προκειμένου να επινοηθεί/καταρτιστεί βήμα προς βήμα πλάνο δράσης (algorithms)

DECOMPOSITION



Pattern Recognition



ABSTRACTION

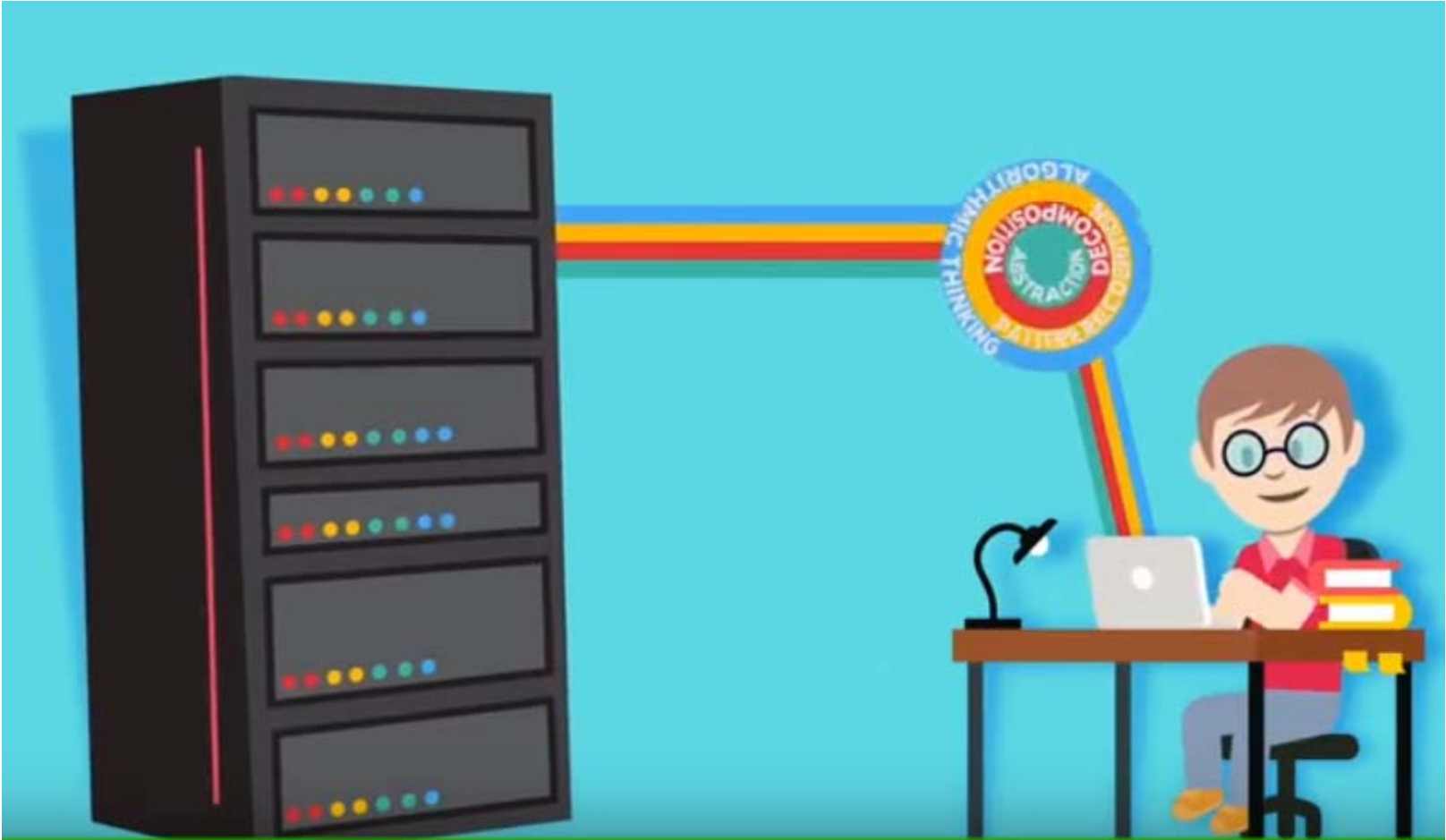


ALGORITHMIC THINKING









Find the sum of all numbers between 1 and 200

$$1 + 2 = 3$$

$$3 + 4 = 7$$

$$5 + 6 = 11$$

$$7 + 8 = 15$$



Find the sum of all numbers between 1 and 200

$$200 + 1 = 201$$

$$199 + 2 = 201$$

$$198 + 3 = 201$$

....

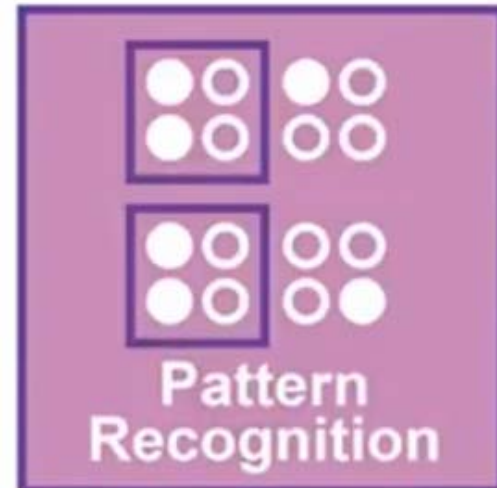
Find the sum of all numbers between 1 and 200

$$200 + 1 = 201$$

$$199 + 2 = 201$$

$$198 + 3 = 201$$

....



Find the sum of all numbers between 1 and 200

$$200 + 1 = 201$$

$$199 + 2 = 201$$

$$198 + 3 = 201$$

....

Find the sum of all numbers between 1 and 200

$$200 + 1 = 201 \quad 200 / 2 = 100 \text{ Pairs}$$

$$199 + 2 = 201$$

$$198 + 3 = 201$$

....

Find the sum of all numbers between 1 and 200

$$\underbrace{(200 + 1)}_{\text{sum of each pair}} * \underbrace{(200 / 2)}_{\text{number of pairs}} = 20,100$$



Abstraction

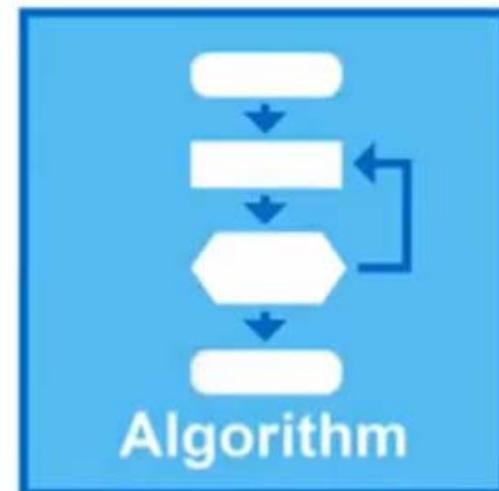
Find the sum of all numbers between 1 and 200

$$(200 + 1) * (200 / 2) = 20,100$$

STEP 1: $200 + 1 = 201$

STEP 2: $200 / 2 = 100$

STEP 3: $201 * 100 = 20,100$



Find the sum of all numbers between 1 and 1,000

$$(1,000 + 1) * (1,000 / 2) = 500,500$$

STEP 1: $1,000 + 1 = 1,001$

STEP 2: $1,000 / 2 = 500$

STEP 3: $1,001 * 500 = 500,500$

Computational Thinking

