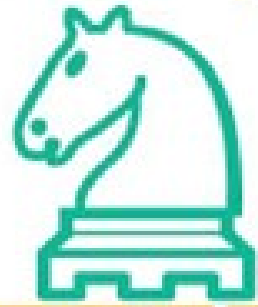


Trainers Support



Algorithmic
Thinking

Algorithmic Thinking for Migrants Teachers Education

2021-1-EL01-KA210-ADU-000035033

ΜΑΘΗΜΑ #3

ΤΙΤΛΟΣ: ΤΕΧΝΙΚΕΣ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ

ΑΠΑΙΤΗΣΕΙΣ ΜΑΘΗΜΑΤΟΣ



ΟΜΑΔΑ: 15
ΕΚΠΑΙΔΕΥΟΜΕΝΟΙ



ΔΙΑΡΚΕΙΑ: 75 ΛΕΠΤΑ



ΠΡΟΒΟΛΕΑΣ, ΥΠΟΛΟΓΙΣΤΗΣ, ΦΥΛΛΟ
ΕΡΩΤΗΣΕΩΝ

ΣΤΟΧΟΙ

1. ΝΑ ΜΑΘΕΤΕ ΧΡΗΣΙΜΕΣ ΤΕΧΝΙΚΕΣ ΓΙΑ ΤΗΝ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ
2. ΝΑ ΕΞΗΓΗΣΕΤΕ ΠΩΣ ΝΑ ΕΦΑΡΜΟΣΕΤΕ ΜΙΑ ΣΥΣΤΗΜΑΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΣΤΗΝ ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΩΝ.
3. ΝΑ ΣΥΖΗΤΗΣΕΤΕ ΠΩΣ ΝΑ ΔΗΜΙΟΥΡΓΗΣΕΤΕ ΕΝΑΝ ΟΡΙΣΜΟ ΠΡΟΒΛΗΜΑΤΟΣ.
4. ΕΙΣΑΓΩΓΗ ΣΤΡΑΤΗΓΙΚΩΝ ΚΑΙ ΠΡΟΒΛΗΜΑΤΙΣΜΩΝ ΓΙΑ ΤΗΝ ΕΞΕΥΡΕΣΗ ΛΥΣΕΩΝ.
5. ΝΑ ΕΞΗΓΗΣΕΤΕ ΤΗΝ ΑΠΟΣΥΝΘΕΣΗ ΩΣ ΣΤΡΑΤΗΓΙΚΗ ΕΠΙΛΥΣΗΣ ΠΡΟΒΛΗΜΑΤΩΝ.

ΟΡΙΣΜΟΣ ΕΝΟΣ ΠΡΟΒΛΗΜΑΤΟΣ

Το πιο δύσκολο μέρος της επίλυσης προβλημάτων είναι ο χαρακτηρισμός του προβλήματος.

Λόγοι για την ανάλυση ενός προβλήματος:

1. Μπορεί να βρούμε πράγματα ανεπιθύμητα για διάφορους λόγους.
2. Ίσως η τρέχουσα διαδικασία είναι πολύ αργή.
3. Ίσως πρέπει τακτικά να λαμβάνουμε ορισμένες αποφάσεις, αλλά έχουμε πάρα πολλά δεδομένα να χειριστούμε.
4. Ίσως έχουμε ελλιπείς πληροφορίες για το πώς συμπεριφέρεται κάτι.

Είναι ανόητο να απαντήσετε σε μια ερώτηση που δεν καταλαβαίνετε:

1. Εάν κάποιος άλλος σας το έδωσε, δοκιμάστε να επαναλάβετε το πρόβλημα με τα δικά σας λόγια.
2. Προσπαθήστε να αναπαραστήσετε το πρόβλημα χρησιμοποιώντας εικόνες και διαγράμματα. Οι άνθρωποι αντιμετωπίζουν καλύτερα τις οπτικές αναπαραστάσεις.
3. Θα υπάρχουν γνωστοί και άγνωστοι στην αρχή. Θα πρέπει να βεβαιωθείτε ότι είναι γνωστές αρκετές πληροφορίες για να διαμορφώσετε μια λύση. Εάν δεν υπάρχει, κάντε τα άγνωστα σαφή



ΟΡΙΣΜΟΣ ΕΝΟΣ ΠΡΟΒΛΗΜΑΤΟΣ

- Όποιο και αν είναι το πρόβλημα, το βασικό πράγμα που πρέπει να θυμόμαστε είναι ότι ένας στόχος καθορίζει τι πρέπει να γίνει και όχι πώς πρέπει να γίνει, σκεπτόμενοι λεπτομέρειες όπως σχέδια και αλγόριθμοι. Εστιάστε αντίθετα στο πώς φαίνεται ο στόχος σας.
- Όπως κι αν καθορίσετε τον στόχο, βεβαιωθείτε ότι η γλώσσα σας είναι σαφής και συγκεκριμένη. Για παράδειγμα, εάν σκοπεύετε να βελτιώσετε την ταχύτητα του τρέχοντος συστήματος, μην καθορίσετε τον τελικό στόχο σας απλώς ως «θα πρέπει να είναι ταχύτερος». Δώστε του μετρήσιμη ακρίβεια

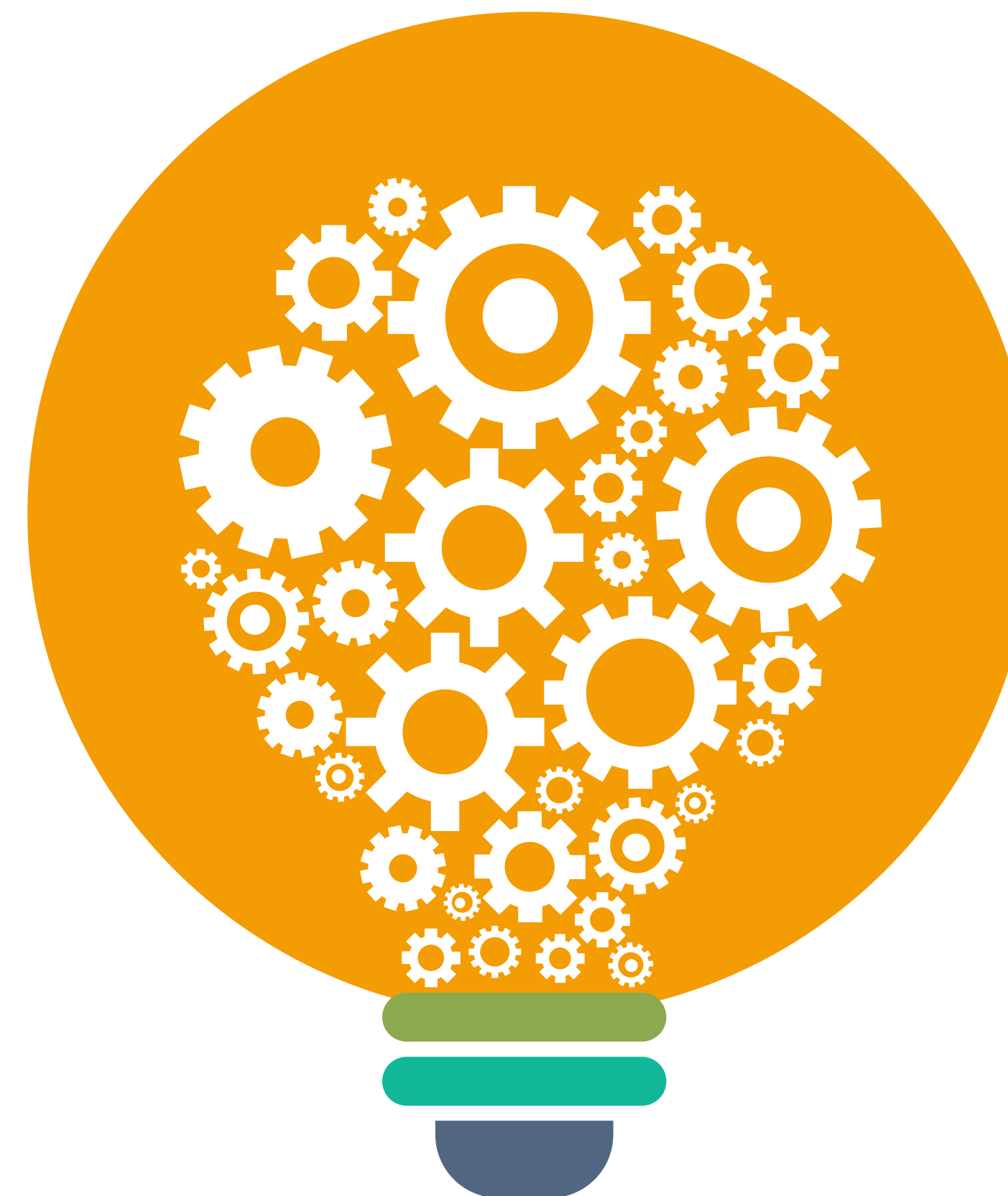


ΕΠΙΝΟΩΝΤΑΣ ΜΙΑ ΛΥΣΗ: ΚΑΤΙ ΠΟΥ ΠΡΕΠΕΙ ΝΑ ΕΧΕΤΕ ΚΑΤΑ ΝΟΥ

1. Ποιότητα

Πρώτον, παρατηρήστε ότι λέμε μια λύση και όχι τη λύση. Για οποιοδήποτε πρόβλημα, υπάρχουν συνήθως πολλές λύσεις: μερικές καλές, μερικές τρομερές και άλλες κάπου ενδιάμεσα. Θα πρέπει να επικεντρωθούμε στην εύρεση της καλύτερης δυνατής λύσης.

Για το συνολικό πρόβλημα, πιθανότατα δεν υπάρχει τέλεια λύση. Οι συμβιβασμοί μεταξύ ανταγωνιστικών τμημάτων είναι σχεδόν αναπόφευκτοι. Από την άλλη, μεμονωμένα μέρη ενός προβλήματος μπορεί να «τελειοποιηθούν», στο βαθμό που ο ρόλος τους στη συνολική λύση μπορεί να είναι βελτιστοποιήσιμος.



ΕΠΙΝΟΩΝΤΑΣ ΜΙΑ ΛΥΣΗ: ΚΑΤΙ ΠΟΥ ΠΡΕΠΕΙ ΝΑ ΕΧΕΤΕ ΚΑΤΑ ΝΟΥ

2. Συνεργασία

Το να κάνουμε την επίλυση προβλημάτων μια συλλογική προσπάθεια είναι συχνά χρήσιμο. Κάτι τόσο απλό όσο το να εξηγούμε δυνατά την τρέχουσα εργασία μας συχνά μας βοηθά να εντοπίζουμε λάθη ή πιθανές βελτιώσεις. Αναζητήστε τις απόψεις των άλλων. Το μυαλό των ανθρώπων λειτουργεί με διαφορετικούς τρόπους.

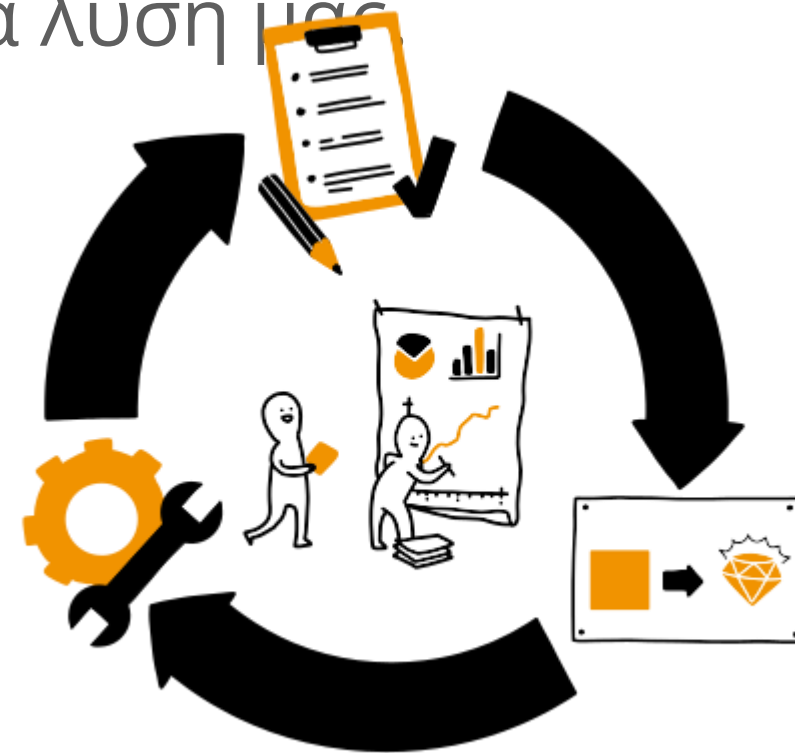
Εφόσον έχετε την προσοχή άλλων ανθρώπων, δοκιμάστε να κάνετε καταιγισμό ιδεών μαζί τους. Οι συνεδρίες καταιγισμού ιδεών ευδοκιμούν στον αυθορμητισμό. Όλες οι ιδέες, όσο ριζοσπαστικές και αν φαίνονται, πρέπει να καταγράφονται και δεν πρέπει να απορρίπτουμε τίποτα εκ προοιμίου. Στην πραγματικότητα, οι ριζοσπαστικές ιδέες πρέπει να ενθαρρύνονται. Ανάμεσα σε αυτές τις τρελές ιδέες μπορεί να βρίσκονται οι σπόροι μιας δημιουργικής νέας προσέγγισης που κανονικά θα μπορούσαμε να χάσουμε ή να αυτολογοκριθούμε.



3. Επανάληψη

Θα πρέπει να δεχτούμε ότι η πρώτη μας προσπάθεια για λύση σπάνια θα είναι η καλύτερη. Αντί να προσπαθείτε να λύσετε τα πάντα με μια κίνηση, ακολουθήστε μια επαναληπτική προσέγγιση.

Επιστρέψτε και επαναλάβετε μερικά από τα προηγούμενα βήματα σε μια προσπάθεια να βελτιώσετε την τρέχουσα λύση μας.

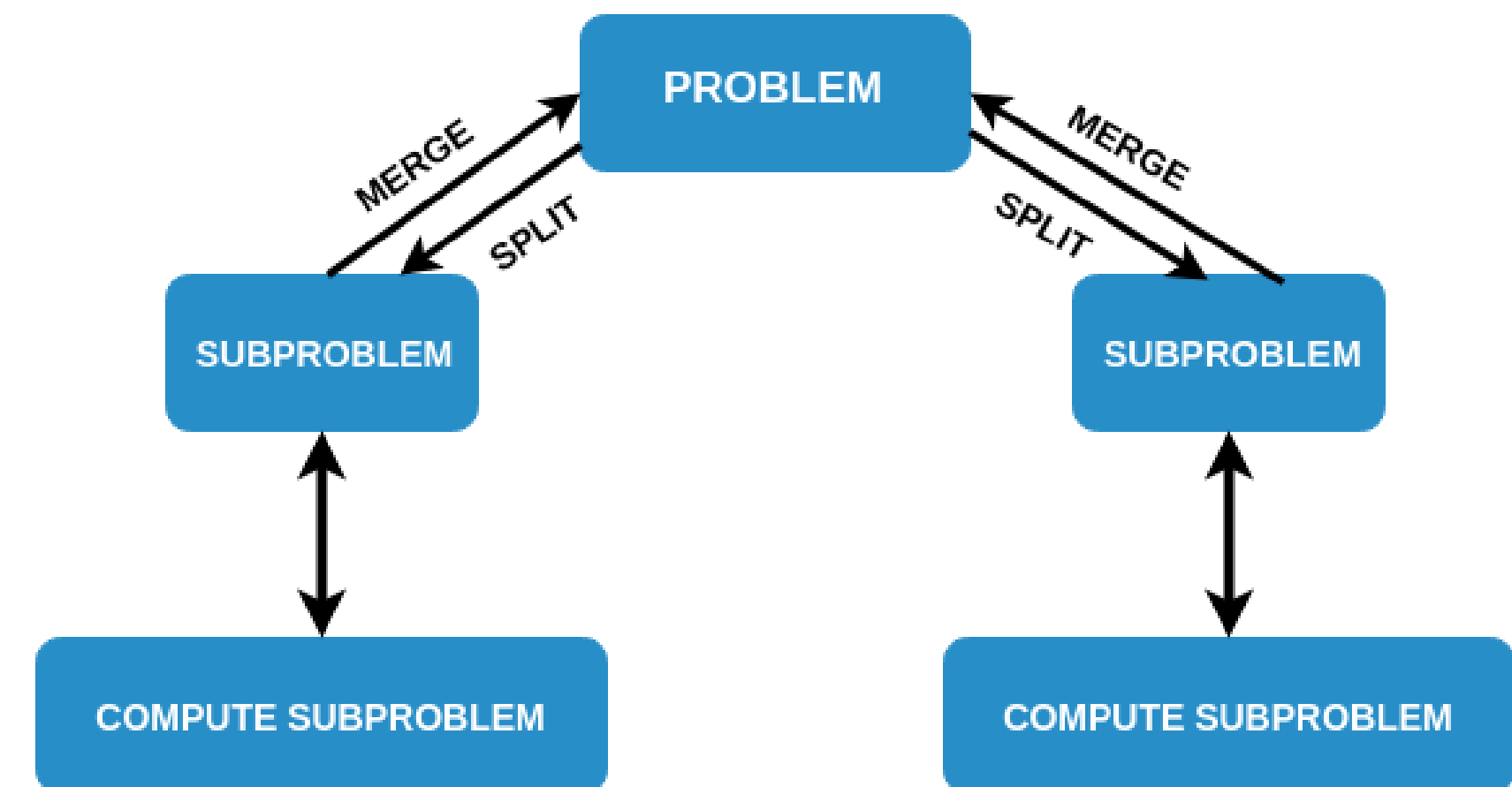


ΑΠΟΣΥΝΘΕΣΗ

Η αλγοριθμική σκέψη προωθεί την αποσύνθεση, η οποία είναι μια προσέγγιση που επιδιώκει να σπάσει ένα σύνθετο πρόβλημα σε απλούστερα μέρη που είναι πιο εύκολο να αντιμετωπιστούν.

Η αποσύνθεση είναι μια στρατηγική διαίρει και βασίλευε, κάτι που παρατηρείται σε πολλά μέρη εκτός υπολογιστών:

1. Οι πολιτικοί το χρησιμοποιούν για να διαλύσουν την αντιπολίτευση σε πιο αδύναμα κόμματα που διαφορετικά θα μπορούσαν να ενωθούν σε ένα ισχυρότερο σύνολο.
2. Όταν αντιμετωπίζουν ένα μεγάλο, διαφορετικό κοινό, οι έμποροι τμηματοποιούν τους πιθανούς πελάτες τους με διαφορετικά στερεότυπα και στοχεύουν τον καθένα διαφορετικά.

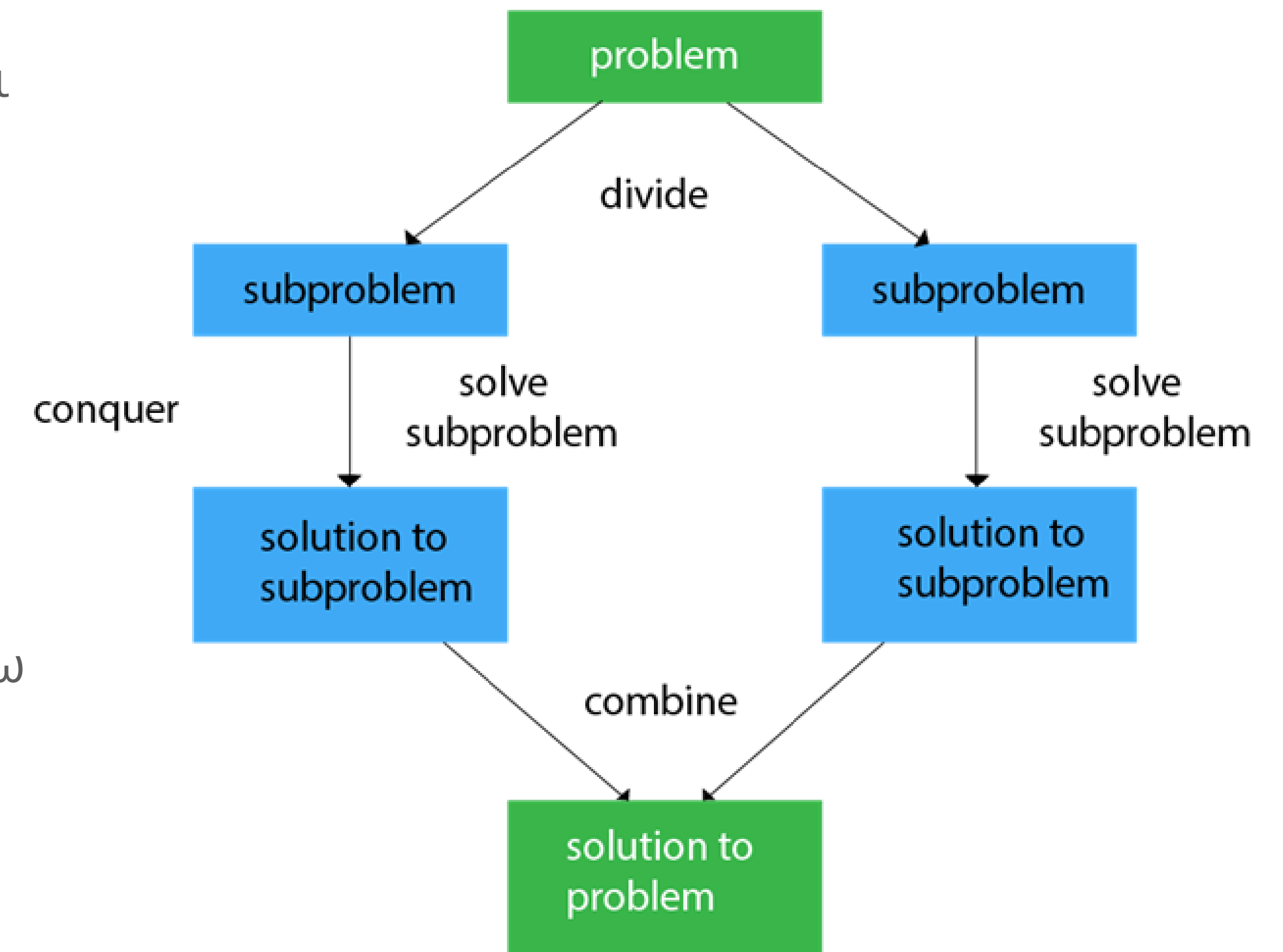


ΑΠΟΣΥΝΘΕΣΗ

Εφαρμόζοντας την αποσύνθεση, στοχεύουμε να καταλήξουμε σε μια σειρά από υπο-προβλήματα που μπορούν να κατανοηθούν και να επιλυθούν μεμονωμένα. Αυτό μπορεί να απαιτεί την εφαρμογή της διαδικασίας.

Δηλαδή, το πρόβλημα ανασχηματίζεται ως μια σειρά μικρότερων προβλημάτων που, ενώ είναι απλούστερα, μπορεί να είναι ακόμα πολύ περίπλοκα, οπότε και αυτά πρέπει να καταρριφθούν και ούτω καθεξής.

Οπτικά αυτό δίνει στον ορισμό του προβλήματος μια δομή δέντρου.



ΔΡΑΣΤΗΡΙΟΤΗΤΑ #3.1

Ο εκπαιδευτής μοιράζει ένα φυλλάδιο με ασκήσεις. Οι ερωτήσεις φαίνονται παρακάτω. Επισημάνετε τις ακόλουθες προτάσεις ως αληθείς ή ψευδείς:

- Ο στόχος καθορίζει πώς πρέπει να λυθεί το πρόβλημα, όχι τι πρέπει να γίνει. (Ψευδής)
- Δεν συνιστάται να αρχίσετε να γράφετε μια λύση πριν καθοριστεί ο στόχος. (Αληθής)
- Για οποιοδήποτε μη τετριμμένο πρόβλημα, πιθανότατα υπάρχει μόνο μία λύση. (Ψευδής)
- Η αποσύνθεση εγγυάται μια βέλτιστη λύση. (Ψευδής)
- Μια δομή δέντρου έχει ιεραρχικό χαρακτήρα. (Αληθής)



ΑΝΑΠΤΥΞΗ ΒΑΣΙΚΩΝ ΔΕΞΙΟΤΗΤΩΝ

- Δεξιότητες επίλυσης προβλημάτων
- Δεξιότητες συλλογισμού
 - Τεχνικές δεξιότητες
- Δεξιότητες διαχείρισης χρόνου

ΧΡΟΝΟΔΙΑΓΡΑΜΜΑ

A
30 λεπτά

ΑΠΑΙΤΟΥΜΕΝΑ ΕΡΓΑΛΕΙΑ

Υπολογιστής, προβολέας, φύλλο χαρτιού

ΑΝΑΦΟΡΕΣ

BEECHER, KARL. 2017. COMPUTATIONAL THINKING: A BEGINNER'S GUIDE TO PROBLEM-SOLVING AND PROGRAMMING. SWINDON, ENGLAND: BCS: THE CHARTERED INSTITUTE FOR IT.

KNUTH, D. (1997) THE ART OF COMPUTER PROGRAMMING, VOLUME 1: FUNDAMENTAL ALGORITHMS. BOSTON, MA, USA: ADDISON-WESLEY

PANE, J. F. ET AL. (2001) STUDYING THE LANGUAGE AND STRUCTURE IN NON-PROGRAMMER'S SOLUTIONS TO PROGRAMMING PROBLEMS. INTERNATIONAL JOURNAL OF HUMAN-COMPUTER STUDIES, 54 (2). 237.

PEA, R. ET AL. (1987) THE BUGGY PATH TO THE DEVELOPMENT OF PROGRAMMING EXPERTISE. FOCUS ON LEARNING PROBLEMS IN MATHEMATICS, 9 (1).