CONCEPTEN

algoritme

WAT?

Een algoritme is een reeks van opeenvolgende instructies of een verzameling regels om een bepaalde taak uit te voeren.

WAAROM?

Bij elke probleem is het belangrijk om dit zo efficiënt mogelijk op te lossen, efficiënt naar tijd, naar materiaal, naar inspanning, naar mogelijkheden ...
Leerlingen kunnen met logisch redeneren fouten in hun eigen algoritmen (of die van anderen) vinden en oplossen, het zogenaamde 'debuggen' of foutopsporing. Het vinden en vervolgens corrigeren van deze fouten stimuleert logisch redeneren en probleemoplossend denken - beide zijn belangrijke computationele denkprocessen.

decompostie

WAT?

Decompositie is het proces waarbij een complex probleem wordt opgedeeld in kleinere deelproblemen. Die aanpak heeft tal van voordelen. Het zorgt ervoor dat we een probleem kunnen beheersen en dat de oplossing veel haalbaarder wordt: grote problemen zijn ontmoedigend, maar een reeks kleinere, gerelateerde taken is veel gemakkelijker aan te pakken. Een groot probleem opdelen in kleinere deelproblemen kan ook samenwerking stimuleren, waarbij elke leerling zijn eigen inzichten, ervaringen en vaardigheden kan inbrengen.

WAAROM?

Eigenlijk is 'decompositie' nu al vaak aanwezig op school. Wanneer leerlingen eender welk project of eender welke taak uitvoeren (zoals een toneelstuk maken, een vraagstuk oplossen, een tekst schrijven of een wafelverkoop organiseren), dan moeten ze die taak opsplitsen in gemakkelijker uit te voeren onderdelen.
Op die manier ontwikkelen ze hun 'decompositie'-vaardigheden.

patroonherkenning

WAT?

Kinderen ontdekken patronen in de wereld om hen heen, zoals dag en nacht, seizoenen, eb en vloed, patronen in muziek, blokkenbouwsels, getalpatronen, kralenketting, vloerpatronen ...
Door patronen in het weer op te merken, verzonnen mensen vroeger weerspreuken, waarna ze die gebruikten als een primitieve vorm van weersvoorspelling.

WAAROM?

We stimuleren leerlingen om patronen op te merken, zodat ze de wereld beter zouden begrijpen.
Computerwetenschappers proberen problemen op een snelle en efficiënte manier op te lossen. De methoden die ze daarvoor gebruiken, willen ze later opnieuw kunnen toepassen. Als ze een gemeenschappelijk patroon zien tussen verschillende problemen of computerprogramma's, dan zullen ze met **éé**n oplossingsmethode komen die ze daarna kunnen hergebruiken. Op die manier hoeven ze het werk maar **éé**n keer te doen.

abstractie

WAT?

Abstractie komt van het Latijnse woord 'abstr**á**here' en betekent 'weglaten'. Komen tot de essentie behoort tot de kern van computationeel denken. Abstractie gaat over zaken eenvoudiger maken, over het identificeren van zaken die belangrijk zijn zonder je veel om details te bekommeren. Abstractie is dus een manier om met complexiteit om te gaan.

WAAROM?

Abstractie is belangrijk om een probleem eenvoudig te kunnen oplossen en om een probleem aan een computer te kunnen voorleggen.
Bij jonge kinderen zijn er al tal van kansen om te abstraheren (vereenvoudigen, weglaten). Je vraagt hen bijvoorbeeld om een gebeurtenis na te vertellen en na te denken welke onderdelen van het verhaal belangrijk zijn en hoe ze het kort kunnen samenvatten. Leerlingen leren zaken te vereenvoudigen en samen te vatten, waardoor ze almaar meer bedreven worden in abstraheren.

debuggen

WAT?

Wanneer algoritmes of programma's niet meer correct functioneren, moeten leerlingen aangemoedigd worden om ze via logisch redeneren te gaan debuggen (foutopsporing). Het is daarbij de bedoeling dat ze over iedere stap logisch gaan nadenken, tot ze bij de stap terechtkomen waarin het fout gaat. De leerlingen kunnen daarbij het algoritme of programma eerst opdelen in kleinere stukjes, zodat ze elk stuk afzonderlijk kunnen testen. Ook hier wordt er verwacht dat de leerlingen nadenken over de meest logische manier om het algoritme of programma op te delen. Eens ze de oorzaak van de fout gevonden hebben, zullen de leerlingen opnieuw logisch moeten redeneren om een oplossing te bedenken.

WAAROM?

Leerlingen kunnen met logisch redeneren fouten in hun eigen algoritmen (of die van anderen) vinden en oplossen = debuggen.
Het vinden en vervolgens corrigeren van deze fouten stimuleert logisch redeneren en probleemoplossend denken. Beide zijn belangrijke computationele denkprocessen.

VAARDIGHEDEN - DEELCOMPETENTIES

automatisering
Leerlingen passen standaardprocedures toe die rechtstreeks naar een oplossing leiden via algoritmisch denken.

omgaan met gegevens
Leerlingen analyseren, verzamelen en visualiseren gegevens.

parallellisme
Leerlingen ontdekken dat een taak sneller klaar is wanneer ze handelingen gelijktijdig uitvoeren.

her)formuleren van problemen
Leerlingen drukken de probleemstelling uit in eigen woorden.

simulatie en modelleren
Leerlingen bootsen de werkelijkheid na.

voorspellen
Leerlingen geven aan wat ze denken dat er zal gebeuren.