



Sorteringsalgoritmer: unplugged

Lektionsoversigt

Målgruppe: 12–16 år. Avanceret niveau.

Varighed: 50 minutter

Læringsmål:

I denne aktivitet lærer eleverne, hvordan nogle almindelige datasorteringsalgoritmer fungerer gennem en unplugged og hands-on tilgang ved brug af en vægt og flere beholdere med forskellig vægt. Formålet er at forstå, hvad der sker, når vi beder om at sortere tv-serier på en streamingplatform, fotos i vores galleri eller produkter i en webshop.

I dette tilfælde bruger vi altid det samme kriterium (vægten af hver beholder), men dette kriterium kunne være hvad som helst (populariteten af tv-serier, tidspunktet for fotos i et galleri eller prisen i en webshop).

Gennem aktiviteten vil eleverne kunne:

- Sortere ethvert datasæt ved hjælp af et simpelt værktøj, der kun kan fortælle os, hvilket element der kommer før eller efter, når to elementer sammenlignes.
- Finde det største eller mindste element i et datasæt.
- Sammenligne fordele og ulemper ved nogle af de mest kendte sorteringsalgoritmer (selection sort, bubble sort, quick sort).

Online or offline: offline

Computational Thinking:

- **CT-grundkompetencer:**
 - Nedbrydning (Decomposition): at følge en “del-og-hersk”-tilgang i rekursive sorteringsalgoritmer som quicksort.
 - Generalisering: at forsøge at bruge den samme tilgang til ethvert sæt af elementer, der skal sorteres.
 - Mønstre (Patterns): at finde relationerne mellem data og muligheden for at sortere dem efter forskellige kriterier.
 - Algoritmisk tænkning: systematisk at gengive trinene i sorteringsalgoritmer.



Materialer

- <Salt / sand / ris (+ 1 ske).
- 7 eller flere uigennemsigtige beholdere.
- Snor / garn.
- Præcisionsvægt (kun til forberedelse, ikke nødvendig senere).
- Legetøjsvægt eller bøjle (fx en tøjbøjle).

Forberedelse

1. Fyld beholderne med forskellige mængder salt/sand/ris. Mål dem med en præcisionsvægt for at sikre, at vægtene er forskellige.



2. Klip et lille stykke snor/garn og luk beholderne, så de nemt kan hænges op i en tøjbøjle.



3. Test tøjbøjlens brug som vægt:



Eller brug en legetøjsvægt (nemmere og sjovere):



Lektionsbeskrivelse – Sorteringsalgoritmer

Byd eleverne velkommen og forklar udfordringen:

“Ved I, hvad der sker, når I klikker på knappen for at sortere produkter efter pris, nyhed eller popularitet i en webshop? Dataene sorteres ved hjælp af en sorteringsalgoritme.

Hvis I havde 10 bøger på jeres bord, hvilke trin ville I tage for at arrangere dem i en pæn stak med den største bog nederst og den mindste øverst? Kan I komme i tanke om andre måder at gøre det på? Er nogle metoder bedre eller dårligere end andre?”

Kerneaktivitet

Opvarmingsaktivitet (10 minutter)

- Opfordr eleverne til at tænke over de forskellige kriterier, vi bruger til at sortere ting: vægt, pris, volumen, farve osv.

Refleksionsspørgsmål

Kan vi kvantificere (tildele et tal til) enhver egenskab ved et objekt? (fx vægt)

Kan vi sortere noget, vi ikke kan kvantificere?

Er det nødvendigt at kende den præcise værdi for hvert objekt, eller er det nok blot at kunne sammenligne to objekter?

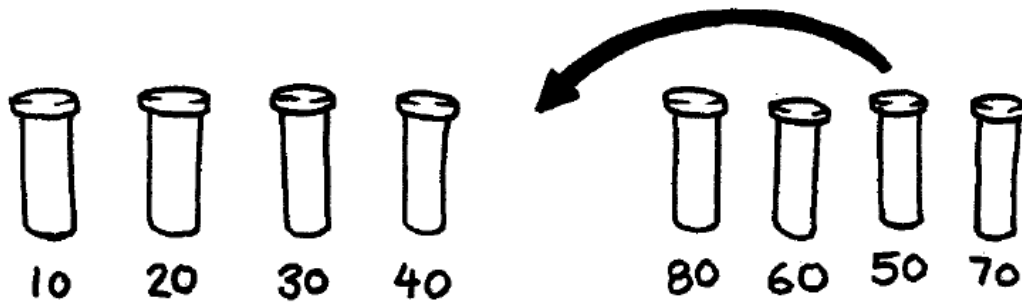
Find eksempler, der understøtter eller modsiger jeres svar.

- Knyt aktiviteten til Computational Thinking:
 - Algoritmisk tænkning: Skriv alle de trin ned, der er nødvendige for at sortere tre objekter efter vægt.
 - Generalisering: Hvad ville der ske, hvis der var fem objekter? Hvad hvis der var 1.000? Ville den samme tilgang stadig virke?
 - Nedbrydning: Kan I finde på en strategi til at opdele et stort sorteringsproblem (fx sortering af 1.000 kort) i mindre dele?



Hovedaktivitet – Sorteringsalgoritmer

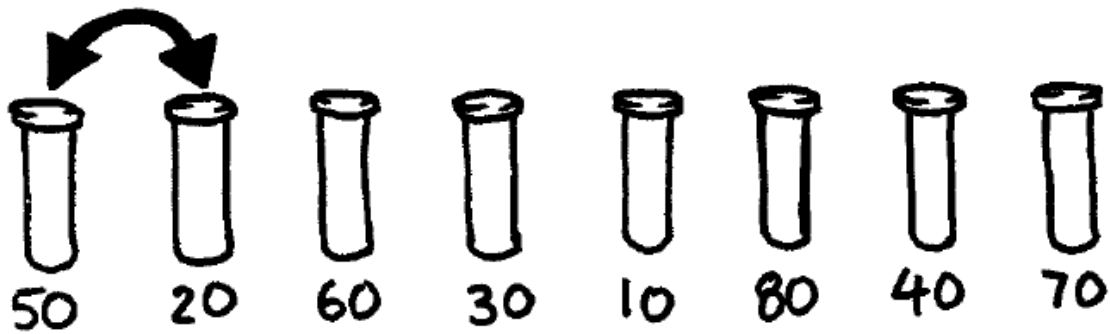
Selection Sort (10 minutter)



- 1. Del eleverne op i grupper (én gruppe per vægt eller tøjbøjle).
- 2. Giv hver gruppe en vægt/bøjle og et sæt beholdere med forskellig vægt.
- 3. Lad grupperne blande beholderne.
- 4. Bed hver gruppe om at følge selection sort-algoritmen:
 - a. Find det letteste element i sættet. Sådan findes det letteste element:
 - Vælg et element som kandidat.
 - Sammenlign det med resten af elementerne.
 - Hver gang et element er lettere end kandidaten, bliver det den nye kandidat.
 - b. Fjern det letteste element fra sættet, og placer det til venstre i et nyt sæt.
 - c. Gentag trinene, indtil der ikke er flere elementer tilbage.
 - d. Når algoritmen er færdig, vil elementerne være sorteret i det nye sæt.



Bubble Sort (10 minutter)

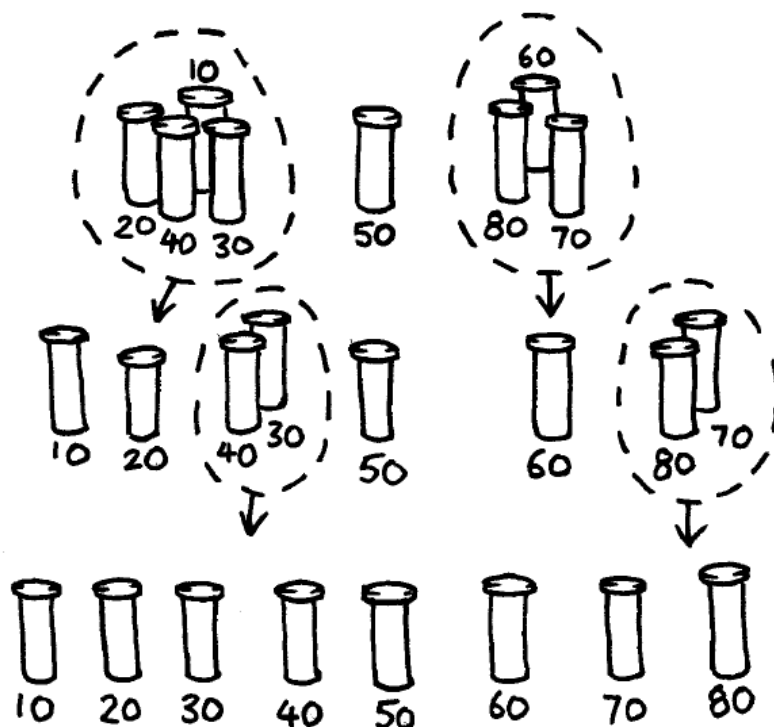


- Del igen eleverne op i grupper, og giv hver gruppe en vægt/bøjle og et sæt beholdere med forskellig vægt.
- Bed grupperne om at følge bubble sort-algoritmen:
 - a. Sammenlign hvert element med det næste og byt dem, hvis de ikke er i korrekt rækkefølge.
 - b. Gentag fra starten, indtil der ikke længere er nogen ændringer.

Refleksion / Diskussion (10 minutter)

- Hvilken algoritme er bedst: selection sort eller bubble sort? Hvorfor?
 - a. Tænk på situationer, hvor den ene er hurtigere end den anden. Hvilke situationer er mest sandsynlige i virkelige datasæt?
- Kan disse algoritmer udnytte en "del-og-hersk"-tilgang?
 - a. Kan flere personer, der sorterer det samme datasæt samtidig, gøre processen hurtigere med disse algoritmer?

Quick Sort (10 minutter)



- Del eleverne op i grupper, giv hver gruppe en vægt/bøjle og et sæt containere, og bland dem.
- Bed grupperne om at følge quick sort-algoritmen:
 - a. Vælg en beholder tilfældigt – dette er drejepunktet (pivot).
 - b. Placer alle beholdere, der er lettere end pivot, til venstre, og alle tungere beholdere til højre.
 - c. Gentag processen for beholderne til venstre og højre.
- Refleksion: Kan denne algoritme drage fordel af en “del-og-hersk”-tilgang? Kan flere personer sortere det samme datasæt hurtigere med denne algoritme?

Opsamling og evaluering (5 minutter)

- Stil refleksionsspørgsmål:
 - Kunne I opfinde en anden sorteringsalgoritme? Tror I, at jeres algoritme er bedre end dem, vi har set, i bestemte tilfælde?
 - Kunne I opfinde en algoritme, som gør et datasæt uordnet? Ville datasættet altid blive uordnet på samme måde?
 - Hvornår tror I, at den slags algoritmer bliver brugt?





Appendix 1

Computer Science Unplugged

<https://www.csunplugged.org/>

Comparison of several sorting algorithms

 Play All	 Insertion	 Selection	 Bubble	 Shell	 Merge	 Heap	 Quick	 Quick3
 Random								
 Nearly Sorted								
 Reversed								
 Few Unique								

<https://www.toptal.com/developers/sorting-algorithms>

