

TIETORAKENTEITA JA TEHOKKUUTTA

Hanna Aarnio ja Virpi Sumu
Resurssikeskus Linkki, HY

MITÄ LINKKI TEKEE?

- Kerhoja ja leirejä lapsille
- Koululuokille vierailuja tietojenkäsittelytieteen laitokselle
- Lukio-opintoja: MOOC:it ja dynamiitit
- **Täydennyskoulutusta, tukea ja materiaaleja opettajille**
- Tapahtumia ja tietoja suurelle yleisölle



Kuvaaja: Arto Wikla

MITÄ ON TEHOKKUUS?

Tehokas = mahdollisimman vähän resursseja vievä.

Tietokoneella siis

- Suoritus aika (prosessoriaika)
- Käytetty muisti
- Muut resurssit, kuten levytila, verkkokaista yms.

Tehokkuuden yhteydessä viitataan tutkittavan alueen kokoon.

Jos vaikkapa sanalistan pituus on n sanaa, listan

läpikäymisen vaativuus on n .

ESIMERKKI: JÄRJESTÄMINEN

Miten
järjestäisit
korttipakan?



Kuva: [Wikipedia](#)

ESIMERKKI: JÄRJESTÄMINEN

Miten järjestäisit korttipakan?

- Ensin pitää päättää, millä kriteereillä järjestetään.
 - Pienemmästä suurempaan, vai toisinpäin?
 - Maat sarjassa, vai numerot sarjassa?
 - Mikä maa on “pienin/suurin”?
- Sitten järjestelytapa:
 - Yksi kerrallaan paikalleen sijoittaen?
 - Pakasta aina seuraavan kortin etsien?
 - Maat ensin kasoihin, sitten numerojärjestykseen?
 - Numerot ensin kasoihin, sitten maat järjestykseen?
 - ...?

ESIMERKKI: JÄRJESTÄMINEN

- Yksi kerrallaan paikalleen sijoittaen?
 - lisäyslajittelu (insertion sort)
- Pakasta aina seuraavan kortin etsien?
 - ~valintalajittelu (selection sort)
- Maat ensin kasoihin, sitten numerojärjestykseen?
- Numerot ensin kasoihin, sitten maat järjestykseen?
 - ...lomituslajittelun (merge sort) suuntaan hiipivää

Entä jos kortteja on 100? 1000? 1 000 000? Mikä on tehokasta ja millä kriteereillä?

ESIMERKKI: JÄRJESTÄMINEN

Järjestysalgoritmeja - n on järjestettävän listan pituus

- Lisäslajittelu: <http://eppabasic.fi/#JMNv1A>
 - Hidas (tyypillisesti/pahimmillaan $n*n$), mutta vie vähän tilaa (1 “muistipaikan”)
- Lomituslajittelu: <https://fi.wikipedia.org/wiki/Lomituslajittelu>
 - Tehokas (tyypillisesti/pahimmillaan $n*\log n$), mutta vaatii lisätilaa (n “muistipaikkaa”)
- Pikalajittelu: <https://fi.wikipedia.org/wiki/Pikalajittelu>
 - Tehokas (tyypillisesti $n*\log n$, pahimmillaan $n*n$), vaatii lisätilaa (tyypillisesti $\log n$, pahimmillaan n “muistipaikkaa”)

ESIMERKKI: JÄRJESTÄMINEN

Harjoituksia järjestämisalgoritmeista:

<http://csunplugged.org/sorting-algorithms/>

Pajassa kokeiltiin lomituslajittelua ja pikalajittelua numeroitujen lappujen avulla.

MIKSI JÄRJESTÄMINEN ON TÄRKEÄÄ?

Esimerkki: puhelinluettelo - mitä jos se ei olisi järjestyksessä?

→ koko puhelinluettelo pitäisi käydä läpi joka kerta

Tietokoneet hakevat tietoa koko ajan, joten asioiden täytyy olla järjestyksessä.

Miten tietoa haetaan esimerkiksi puhelinluettelosta?

TEHOKAS HAKEMINEN

Miten tietoa haetaan esimerkiksi puhelinluettelosta?

- Alusta alkaen sivu tai nippu sivuja kerrallaan?
 - Pahimmillaan pitää selata viimeiselle sivulle asti, eli taas suorituksen vaativuus on $n...$
 - Tietokone ei helposti osaa tehdä intuitiivisia oletuksia tiedon sijainnista!
- Keskeältä alkaen katsotaan kumpaan suuntaan kannattaa mennä? → Binäärihaku (<https://fi.wikipedia.org/wiki/Puolitushaku>)
 - Tutkittava alue puolittuu joka kerta, joten pahimmillaan vaativuus on $\log n$
 - Yleensä siis tehokkain
- Muita tapoja...?

JATKOHAASTEITA

Aloittelijoille OhjelmointiMOOC

<http://mooc.fi/courses/2016/ohjelmoinnin-mooc/>

Kokeneemmille esim. algoritmien AlgoMOOC

<http://mooc.fi/courses/2015/algoritmit/>