

## Lisäesimerkki rekursiosta

- Katsotaan vielä esimerkki muunlaisesta rekursiosta kuin jonkin numerisen funktion laskemisesta
- Tehtävänä on tulostaa jonkin  $n$ -alkioisen perusjoukon kaikki  $k$ -alkioiset osajoukot. Yksinkertaisuuden vuoksi oletetaan, että perusjoukko on  $\{0, \dots, n - 1\}$
- Miten tehtävän voi ratkaista  $n$ -alkioisella perusjoukolla, jos osaamme jo ratkaista sen  $(n - 1)$ -alkioisella?
- Joukon  $\{0, \dots, n - 1\}$  kaikki  $k$ -alkioiset osajoukot (kun  $k > 0$ ) voidaan luetella seuraavasti:
  - Luettele joukon  $\{1, \dots, n - 1\}$  kaikki  $(k - 1)$ -alkioiset osajoukot siten, että niiden eteen on liitetty alkio 0
  - Luettele joukon  $\{1, \dots, n - 1\}$  kaikki  $k$ -alkioiset osajoukot
- Jälkimmäinen luettelo jää tyhjäksi tapauksessa  $k = n$

- Saamme siis  $n$ -alkioisen ongelman suunnilleen palautetuksi  $(n - 1)$ -alkioiseen
- Rekursion toteuttaminen vaatii kuitenkin hieman lisäinformaatiota:
  - mitkä alkiot on jo valittu joukon alkuun
  - mikä on pienin jäljellä oleva alkio
- Kerätään seuraavaksi tulostettava osajoukko taulukkoon  $A$  alkio kerrallaan
- Määritellään rekursiivinen algoritmi **tulostaJoukot**( $p, q$ ), joka tulostaa kaikki sellaiset  $k$ -alkioiset osajoukot, joissa
  - pienimmät  $p$  alkioita ovat  $A[1], \dots, A[p]$
  - loput  $k - p$  alkioita valitaan joukosta  $\{q, \dots, n\}$
- Kutsu **tulostaJoukot**( $0, 0$ ) tuottaa nyt halutun tuloksen

Algoritmi on seuraava:

```
tulostaJoukot( $p, q$ )
1  if  $p < k$ 
2       $A[p + 1] = q$ 
3      tulostaJoukot( $p + 1, q + 1$ )
4      if  $k - p < n - q$ 
5          tulostaJoukot( $p, q + 1$ )
6  else
7      tulosta joukko  $\{ A[1], \dots, A[k] \}$ 
```

- $n$  ja  $k$  oletetaan määritellyksi jossain muualla
- rivin 5 ehtotestin tarkoitus on estää alkion  $q$  jättäminen pois, jos jäljellä on enää täsmälleen tarvittava määrä alkioita