

# Udvælgelse af $k$ 'te mindste element

## Dat2A godkendelsesopgave 1

Jens Kristian Jensen og Martin Zachariasen

25. november 2002

### 1 Formalia

Dette er den første af to godkendelsesopgaver på kurset Dat2A, efterår 2002 og forår 2003. Opgaven skal løses i grupper på 2–3 personer (enkeltmandsgrupper kræver begrundelse og skal godkendes af instruktør). Opgaven bliver stillet tirsdag d. 26/11 2002 kl. 11.30 og skal afleveres i førstedelsadministrationen i 2 eksemplarer inden **fredag d. 6/12 2002 kl. 11.30**. Bedømmelserne foreligger i førstedelsadministrationen senest mandag d. 16/12 2002. I tilfælde af at opgaven ikke bliver godkendt i første omgang er genafleveringsfristen fredag d. 20/12 2002 kl. 11.30.

### 2 Formål

Formålet med opgaven er at

- få et indblik i køretidsanalyser for rekursive algoritmer
- få praktisk erfaring med implementation af (rekursive) algoritmer
- at blive bedre i stand til at relatere teoretiske køretidsanalyser til reelle køretider på virkelige maskiner

Formålet med opgaven er at undersøge en algoritme til udvælgelse af  $k$ 'te mindste element fra en tabel med  $n$  elementer. Algoritmen skal undersøges eksperimentelt, og desuden skal et antal relaterede teoretiske spørgsmål besvares. Til sidst skal algoritmen til udvælgelse af  $k$ 'te mindste element anvendes til at

forbedre den praktiske køretid af Kruskals mindste udspændende træ (MST) algoritme.

## 3 Om opgaven

### 3.1 LEDA

Alle algoritmer og datastrukturer skal implementeres i C++ og gerne ved anvendelse af klassebiblioteket LEDA. Anvendelse af LEDA på DIKUs maskiner er beskrevet på:

<http://www.diku.dk/forskning/algorithms/libraries.html#LEDA>

### 3.2 Lineær-tids algoritme til $k$ 'te mindste element

Lineær-tids algoritmen til udvælgelse af det  $k$ 'te mindste element fra en tabel — som er beskrevet i afsnit 9.3 i CLRS [1] — skal implementeres. Det anbefales at benytte skabelonen `~dat2a/01/select.cc` som udgangspunkt for implementationen. Bemærk specielt det angivne interface til algoritmen, herunder parameteren `GroupSize`, som angiver antallet af elementer i hver gruppe; denne parameter har værdien 5 i algoritmen fra CLRS.

### 3.3 Kruskals MST algoritme

Implementer Kruskals MST algoritme (afsnit 23.2 i CLRS) ved anvendelse af LEDAs grafklasse. Det anbefales at benytte klassen `edge_array` til at gemme kantvægtene. Kopier desuden kanterne til en array af typen `array<edge>`, og benyt metoden `sort()` til sortering af kanterne. (Kopieringen til array er nødvendig til senere brug.)

## 4 Krav til besvarelsen

Rapporten skal være på højst 10 sider, excl. programudskrifter og udskrifter fra kørsler (der også vedlægges). Desuden skal alle kildetekstfiler — herunder en README-fil der beskriver hvordan filerne oversættes til et færdigt program — gøres tilgængelige på en 1. delskonto (di-konto) ejet af en af deltagerne; filerne lægges i et katalog med navnet `012A` og må ikke rettes efter rapportens aflevering.

I rapporten skal følgende spørgsmål besvares:

1. Lineær-tids udvælgelse: Vis at køretiden af algoritmen er  $O(n)$  for enhver ulige konstant gruppestørrelse større end eller lig med 5.
2. Lineær-tids udvælgelse: Udfør eksperimenter med forskellige værdier af gruppestørrelsen og forklar resultaterne (rækken af værdier skal indeholde 3, 5 og 7). Vælg på baggrund af disse eksperimenter en god værdi for gruppestørrelsen for  $n \approx 500000$ .
3. Kruskals MST algoritme: Vis hvordan udvælgelse af  $k$ 'te mindste element kan benyttes til at forbedre den praktiske køretid af Kruskals MST algoritme. (Vink: Sorter en delmængde af kanterne først, og overvej om det er nødvendigt at sortere alle de resterende kanter.)  
Implementer denne forbedring og sammenlign eksperimentelt den forbedrede algoritme med den originale algoritme. Graferne til eksperimenterne kan eksempelvis genereres som vist i `~dat2a/01/select.cc`.
4. Generaliseret udvælgelse: Lad  $K \geq 1$  være givet. Vi ønsker nu at dele tabellen op i  $2^K$  lige store grupper<sup>1</sup>, så alle elementer i den  $i$ 'te gruppe er mindre eller lig med alle elementer i den  $(i + 1)$ 'te gruppe. Vis at den generaliserede udvælgelse kan foretages i  $O(nK)$  tid.

## 5 Gode råd

Der er spørgetime om opgaven i forbindelse med forelæsningen d. 28/11. Desuden svarer instruktør på spørgsmål i forbindelse med de sædvanlige øvelsestimer. Svar på spørgsmål af generel interesse vil løbende blive lagt ud på kursets hjemmeside.

## Litteratur

- [1] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein: Introduction to Algorithms (2nd ed.), MIT Press, 2001.

---

<sup>1</sup>Gruppernes størrelsesforskel er på max. 1 element.