

58053-7 Algoritmien suunnittelu ja analyysi (kevät 2004)

Harjoitus 8 (18.–19. maaliskuuta)

1. Hahmottele simuloitua jäähtytystä käyttävä ratkaisumenetelmä NP-täydelliselle solmupeiteongelmalle (VERTEX COVER). Valitse sopiva kustannusfunktio ja symmetrinen, tasa-asteinen naapurustorakenne. Varmista, että osaat tuottaa naapurustorakenteen mukaisia satunnaisia naapureita tasaisen jakauman mukaan. Minkälaista jäähtytysaikataulua ongelman ratkaisemiseen n solmun verkossa tulisi käyttää luennolla esitetyn konvergenssilauseen mukaan?
2. (a) Esitä pseudokoodina toteutus binomikasojen yhdistämisessä käytetylle alirutiinille BINOMIAL-HEAP-MERGE.
(b) Simuloi yhdistämisoperaatiota BINOMIAL-HEAP-UNION(H_1, H_2) (luennot s. 216) itse valitsemillasi binomikasoilla H_1 ja H_2 . Valitse H_1 ja H_2 siten, että algoritmin kaikki haarat tulevat käytetyksi.
3. (a) Osoita, että n -solmuiseen binomikasaan kohdistuvat operaatiot BINOMIAL-HEAP-INSERT, BINOMIAL-HEAP-MINIMUM, BINOMIAL-HEAP-EXTRACT-MIN, BINOMIAL-HEAP-UNION, BINOMIAL-HEAP-DECREASE-KEY ja BINOMIAL-HEAP-DELETE voivat vaatia ajan $\Omega(\log n)$.
(b) Joillekin näistä operaatioista on kuitenkin olemassa mielivaltaisen suuria n , joilla *tasan* n -solmuiseen kasaan kohdistuvat operaatiot toimivat vakioajassa. Mitkä operaatiot nämä ovat? Minkä tyyppisistä n on kysymys?
4. Tarkastellaan seuraavaa periaatealgoritmia pienimmän virittävän puun löytämiseksi:

```
MST( $V, E$ ):  
   $T := \emptyset$   
  olkoon  $V = \{v_1, \dots, v_n\}$   
  for  $i := 1$  to  $n$  do  
     $V_i := \{v_i\}$   
     $E_i := \{(u, v) \in E \mid u = v_i \text{ tai } v = v_i\}$   
  while joukkoja  $V_i$  on enemmän kuin yksi do  
    valitse jokin joukko  $V_i$   
    valitse painoltaan pienin kaari  $(u, v) \in E_i$  ja poista se  
    oletetaan  $u \in V_i$  ja  $v \in V_j$   
    if  $i \neq j$  then  
       $T := T \cup \{(u, v)\}$   
       $V_i := V_i \cup V_j$ ; tuhoa  $V_j$   
       $E_i := E_i \cup E_j$ ; tuhoa  $E_j$ 
```

Miten toteuttaisit algoritmin binomikasoja käyttäen? Esitä mahdolliset lisäykset ja muutokset, joita tarvitset tietorakenteeseen tai operaatioihin. Mikä on algoritmin aikavaativuus?

5. Todista, että edellisen tehtävän algoritmi aina todella löytää pienimmän virittävän puun.