Institut for Matematik og Datalogi
Syddansk Universitet February 22, 2010
JFB

DM508 – Algorithms and Complexity – 2010
Lecture 7

Lecture, February 22

We finished NP-Completeness, covering SUBSET-SUM and some general remarks, and began on amortized analysis, covering the first three sections from chapter 17 of the textbook.

Lecture, February 24

We will cover Fibonacci heaps from chapter 20 (chapter 19 in the third edition).

Lecture, March 1

We will begin on string matching from chapter 32.

Problems to be discussed on March 3

Do problems:

1. 20.3-1.
2. 20.4-1.
3. 20-1, 20-2a.
4. Consider the set-up from Problem 1 on the second lecture note, with a company with two service representatives, originally in Detroit and San Francisco, with requests for service coming from Detroit, San Francisco and Chicago. Traveling between Chicago and Detroit costs f, while traveling between Chicago and San Francisco costs df. The company wants to minimize its costs.

   Consider the following algorithm, *Savings*: The representative in San Francisco, called "Rep A", keeps a savings account to pay for tickets, as does the representative in Detroit, called "Rep B". These accounts start at zero. Given a request, if a service representative is already there, do nothing (except empty the account for the representative there if the request was San Francisco or Detroit). If not, and the request is not for Chicago, move the representative in Chicago there (and empty the account for that representative). If the request is for Chicago and no representative there, if the amount in Rep A's account is less than or equal to $(d-1)f$, then move the representative from Detroit to Chicago and have Rep A add f minus the amount in Rep B's account to its savings account and Rep B empty its account. Otherwise, move the representative from San Francisco to Chicago and have Rep B add df minus the amount that was in Rep A's account to its account and Rep A empty its account.

The following questions are an analysis of *Savings* compared to any other algorithm O using amortized analysis. Let $S_A(L)$ be the savings that Rep A has in its account after processing the requests in L and $S_B(L)$ be the same for Rep B. Let the potential function be $\Phi(L) = 2M(L) + C(L)$, where $M(L)$ is the cost that would be necessary (after both *Savings* and O process L) to move *Saving*'s representatives to the same places as O's (minus $S_A(L)$ if Rep A would have to move and plus $S_A(L)$ otherwise, minus $S_B(L)$ if Rep B would have to move and plus $S_B(L)$ otherwise), and $C(L)$ is the cost to fly between the two cities where *Savings* currently has representatives.

Note that the savings accounts have value zero if the representative is in Chicago or if the representative has just returned to its home city. If the one algorithm has representatives in Chicago and San Francisco, while the other has representatives in Chicago and Detroit, we say that to move the representatives to the same places, both servers move (Rep A should never go to Detroit and Rep B should never go to San Francisco.)

(a) Suppose *Savings* has representatives in Chicago and San Francisco. What is the actual cost of servicing a request in Detroit, and what is the amortized cost (consider each of the possible cases of where O could originally have had its representatives)?

(b) Suppose *Savings* has representatives in Chicago and Detroit. What is the actual cost of servicing a request in San Francisco, and what is the amortized cost (consider all possible cases)?

(c) Suppose *Savings* has representatives in Detroit and San Francisco and Rep A has $S_A(L)$ in its savings account. What is the actual cost of servicing a request in Chicago, and what is the amortized cost (consider all possible cases)?

(d) Argue that on any sequence of requests *Savings* has cost at most twice that of any other algorithm plus an additive constant of at most df. (Remember to take into account potential changes when none of *Savings*'s representatives moves.)