Communications and Transport Systems
Department of Science and Technology **Fall 2018**
Linköping University

Christiane Schmidt

Design and Analysis of Algorithms Part 2 - Approximation and online algorithms homework 6, 19.12.2018

Problem 1 (The Ski Rental Problem):
Let r be the ratio of purchasing price (p) to rental fee (f) ($r = \frac{p}{f}$).

a) Give a strategy that achieves a competitive factor of $2 - \frac{1}{r}$. Prove your claim!

b) Let $p = 100$, $f = 20$:

(1) Which competitve factor achieves the stratgey of renting skis for three days and buying skis on day four?

(2) Which strategy achieves a competitive factor of only 3?

Problem 2 (Paging II):
Consider the LIFO strategy.

(a) Is LIFO a marking algorithm?

(b) Is LIFO conservative?

(c) Is LIFO competitive?

Problem 3 (Paging III):
Show that FIFO does incur the Belady's anomaly.

Problem 4 (Paging IV):
Prove: LRU is a marking algorithm (Lemma 6.7 from the lecture).

Problem 5 (Paging V):
Show that algorithm FIFO is not a marking algorithm.

Problem 6 (k-Server Problem):
We consider the k-server problem: an algorithm can move k mobile servers in the plane, in the beginning they are located on points from a set M. Given: a sequence $\sigma = r_1, r_2, ..., r_n$ of requests, where each request is a point in the plane. A request r_i

is served when one of the servers is on r_i. By moving the servers, the algorithm must process the requests in the given order. The cost of the algorithm is the distance covered by all servers (w.r.t. a metric).

Show that a greedy strategy for the k-server problem is not necessarily competitive. A greedy strategy choosed the cost-minimal option in each step, that is, here moving the server that is closest to the requested point.

Hint: Consider $k = 2$ servers and three well-chosen starting points (and an appropriate infinite request sequence.

Problem 7 (Memory):
We consider a version of memory for a single player. Like in the well-known version, we have n pairs of cards on the table, back side up. The player may turn around two cards in each move. If they are a pair, the two cards are removed. Otherwise, they are turned around again.
The player aims to remove all cards in as few moves as possible. We also count turning around two new cards as a move after successfully putting a pair up.
We assume that the player can remember all already seen cards at all times. The optimal offline player needs n moves to remove all $2n$ cards.
Give a 2-competitive strategy and prove its competitive ratio!

Back to approximation algorithms:

Problem 8 (Set Cover):
Take a set cover problem with n elements. Give an approximation algorithm with performance guarantee $ln(\frac{n}{k}) + 1$, where k is the cardinality of the optimal solution.

Problem 9 (Steiner Tree Problem):
Given a graph G with edge costs and set $T \subseteq V$ of terminal vertices. The steiner tree problem is to find a minimum cost tree in G containing every vertex in T (vertices in $V \setminus T$ may or may not be used). Give a 2-approximation algorithm if the edge costs satisfy the triangle inequality.

And one final reduction:

Problem 10 (Graph Coloring (from 3SAT)):
A k-coloring of a graph is a map $C : V \to \{1, 2, \dots, k\}$ that assigns one of k colors to each vertex, so that every edge has two different colors at its endpoints. The graph coloring problem is to find the smallest possible number of colors in a legal coloring. To show that this problem is NP-hard, its enough to consider the special case 3COLORABLE: Given a graph, does it have a 3-coloring? Prove that 3COLORABLE is NP-hard by a reduction from 3SAT.