CSE 202 Homework 2
Winter, 2014
Path search, using and modifying algorithms, fitting data structures to
algorithms. All parts are worth 20 points. Due Tuesday, January 28

Matrix sizes You can only multiply \(r_1 \times c_1\) and \(r_2 \times c_2\) dimensional matrices
if \(c_1 = r_2\), and the result is a \(r_1 \times c_2\) dimensional matrix. Say that you are
given a list of pairs of integers representing dimensions of matrix variables
\(M_1, \ldots M_n\), where \(M_i\) is a \(r_i \times c_i\) dimensional matrix. You want to find a
list of all possible dimensions \((r, c)\) of products of sequences of matrices
from among \(M_1, \ldots M_n\). Give an efficient algorithm for this problem. (10
points correct algorithm, 10 points efficiency).

Flight scheduling You are devising a flight scheduler for a travel agency. The
scheduler will get a list of available flights, and the customer’s origin and
destination. For each flight, it is given the cities and times of departure
and arrival. The scheduler should output a list of flights that will take
the customer from her origin to her destination that arrives as early as
possible, subject to giving her at least 15 minutes for each connection.
Give a formal specification for this problem (Instance, Solution Space,
Constraints, Objective). (5 points) Then give as efficient as possible an
algorithm to solve the problem. (5 points correct algorithm, 10 points
efficiency).

Palindromic path You are given a directed graph \(G\) where every edge \(e\) has
a label, \(l(e) \in \Sigma\), where \(\Sigma\) is a finite set of symbols, and two nodes \(s\) and
\(t\). You want to determine whether there is a path \(e_1, \ldots e_k\) (not necessarily
simple) from \(s\) to \(t\) so that the labels of the edges form a palindrome, i.e.,
\(l(e_1) \ldots l(e_k) = l(e_k) \ldots l(e_1)\). (10 points correct algorithm and correctness
proof, 10 points efficiency.)

Top \(k\) elements in heap: 20 pts Give an efficient algorithm that, given a
binary max-heap \(H\) of size \(n\) and a number \(1 \leq k \leq n\), returns the
\(k\) largest elements of \(H\). Analyze the time in terms of both \(k\) and \(n\)
(although, for the best algorithm I know, the time does not depend on \(n\)
at all, just \(k\)).
(5 points correct algorithm, 15 points efficiency).

Implementation problem: Popular Websites A web-company wants a data
structure that will display webpages by popularity, displaying the most
popular. The input will be a sequence of IP addresses. Intermittently,
the data structure will need to display the top \(k\) most frequently visited
sites (where \(k\) is given by the user). The data structure should update its
list after every new site hit. So the data structure needs to store a list of
webpages, ordered by \(hit(site)\), the number of hits on the site. It needs to
update this list each time a new hit is made, i.e., Update(site) adds site
to the list with 1 hit, if it isn’t already there, or increments hit(site) if
it is. Top(k) needs to report the top k most popular sites. Describe and
implement at least two data structures for this problem. Compare their
performances on test data generated as follows:

Test distribution: A sequence of 1,000,000 random web addresses, each
with probability 1/4 of being of the form: random 3 letter word.cs.edu
probability 1/4 of the form: free.random 3 letter word.com and probability
1/2 of the form: random 2 letter word.random 2 letter word.com, org, edu.
All words are lower case and have only standard letters. After every 1000
sites, perform Top(k) for \( k = 2^i \), i uniformly chosen from 0 to 10.

Discuss any conclusions or issues that arose.