Assignment 1

Dynamic Programming and the Greedy Algorithm

COMP-362 Winter Term 2004

1. Using dynamic programming, give an algorithm that solves the Traveling Salesman Problem. That is, given a collection of n cities and a cost matrix C, the algorithm will be returning a shortest tour from city 1 to itself.

   Use a pseudo-code, but be as explicit as possible in terms of the data structures you will use.

2. Given n matrices $A_1, \ldots, A_n$ compute the total number of parenthesizations. Give (and prove) an asymptotic expression for this. More specifically, if $P(n)$ is the total number of parenthesizations, express $P(n) = f(n)(1 + o(1))$, for some function $f(n) : \mathbb{N} \to \mathbb{R}$ (apart from the trivial one $P(n) = P(n)(1 + o(1))...$).

3. We are given a sequence of natural numbers $a_1 \ldots a_n$ and consider all possible ways of partitioning them into m parts (How many such ways are there?). Each partition has a cost calculated as follows: if the partition is $a_1 \ldots a_{i_1} | a_{i_1+1} \ldots a_{i_2} | \cdots | a_{i_{m-2}+1} \ldots a_{i_{m-1}} | a_{i_{m-1}+1} \ldots a_n$, then its cost is $(a_1 + \cdots + a_{i_1})^3 + (a_{i_1+1} + \cdots + a_{i_2})^3 + \cdots + (a_{i_{m-1}+1} + \cdots + a_n)^3$.

   Using dynamic programming, give an algorithm that returns a partition of minimum cost. Estimate that time and space requirements.

4. Let S, T be two finite and non-empty sets and let $R \subseteq S \times T$ be a binary relation. Suppose that each element of S has a non-negative weight, that is there is a function $w : S \to \mathbb{R}_+$. Also, for a set $A \subseteq S$, we let $N(A) = \{y \in T \ : \ \exists x \in A \text{ with } (x, y) \in R\}$.

   A set A is called *matchable into* T if $\forall B \subseteq A$ we have $|B| \leq |N(B)|$. The relation R can be seen as a bipartite graph $G = (S \cup T, R)$. A *matching* in a graph is a subset of its edges so that no two of them have end-points in common. Hall's Theorem (which we will see later in the lectures) characterizes the subsets of S that are matchable into T: a set $A \subseteq S$ is matchable into T if and only if there is a matching $M \subseteq R$ in the corresponding bipartite graph such that the end-vertices of the edges in M that belong to S are the elements of A.

   Give a greedy algorithm which, given two finite non-empty and disjoint sets S, T, a weight function w on S and a binary relation $R \subseteq S \times T$, returns a subset of S that is matchable into T and has maximum weight.

Due date: 27th January 2004