Technische Universität München
Department of Computer Science
Chair for Efficient Algorithms
Prof. Dr. Sami Khuri

Summer semester 2000
Problem Set 10
July 4, 2000

Efficient Algorithms and Data Structures II

Problem 1 (10 credits)

Consider the following game, called *The Game of Erase*. You are given a string of length u with arbitrary characters. The only legal moves you are allowed to make are as follows.

Type 1. Erase a single character from the string.

Type 2. Erase two characters from the string if they are identical and adjacent.

The goal of the game is to find a minimal sequence of moves that will erase the whole string. For example, if the input string is **AACBC**, then the following sequence of moves is minimal: **AACBC** $\rightarrow$ **CBC** $\rightarrow$ **CC** $\rightarrow \epsilon$, where ϵ denotes the empty string.

For each statement below, determine, with proof, whether it is true or false.

a. If, for a given string $S = s_1 \cdots s_u$ of length u, the length of a minimal sequence of moves is n, then there exists a minimal sequence M that erases S such that one of the following is true.

1. The first move of M erases s_1, and the other $n-1$ moves erase $s' = s_2 \cdots s_u$ using a minimal sequence for S'.
2. For some index ℓ, $1 < \ell \leq u$, $s_1 = s_\ell$, and for some i, $0 \leq i < n$, the first i moves of M erase the substring $S' = s_2 \cdots s_{\ell-1}$ using a minimal sequence for S', the $(i+1)$-st move of M erases s_1 and s_ℓ, and the remaining $n-i-1$ moves of M erase the substring $S'' = s_{\ell+1} \cdots s_u$ using a minimal sequence for S''.

b. For all i, if there is a minimal sequence M of n moves that deletes s_i alone, i.e., using **type 1** move above, then for any $1 \leq j \leq n$, there exists a minimal sequence of moves for which deleting s_i is the j-th move.

c. For all $1 \leq i \leq u$, if there is a minimal sequence M of n moves that deletes s_i alone, then there exists a minimal sequence of moves M' such that M' first deletes $s_1 \cdots s_{i-1}$, then deletes s_i, and then deletes $s_{i+1} \cdots s_u$.

d. Let $p_{i,j}(S)$ denote the length of a minimal sequence of moves that erases the substring $s_i \cdots s_j$. If $j < i$, $p_{i,j}$ is defined to be 0. Then,

$$p_{i,j}(S) = \min\{1 + p_{i+1,j}(S), \min_{1<\ell\leq j,\ s_i=s_\ell}\{1 + p_{i+1,\ell-1}(S) + p_{\ell+1,j}(S)\}\}.$$

Turn over.

Problem 2 (10 credits)

Let $G = (V, E)$ be a weighted, undirected graph. The *3Max-Cut* problem consists of partitioning V into three pairwise disjoint subsets V_1, V_2, and V_3, called clusters, such that the total cost of edges between clusters, called cross edges, is maximized. Consider the following approximation algorithm. Does APPROX-3MAX-CUT have a ratio bound? Why?

```
APPROX-3MAX-CUT(G)
    V1, V2, V3 ← ∅
    cluster, sum ← 0
    for each u ∈ V do
        mincost ← ∞
        for i ← 1 to 3 do
            cost ← 0
            for each v ∈ V_i do
                if (u, v) ∈ E then cost ← cost+c(u, v) endif
            endfor
            sum ← sum + cost
            if cost < mincost then
                mincost ← cost
                k ← i
            endif
        endfor
        V_k ← V_k ∪ {u}
        cluster ← cluster + mincost
        cross ← sum − cluster
    endfor
    return cross
```

Problem 3 (10 credits)

Multiprocessor Scheduling: Consider the simplest version of the multiprocessor scheduling problem where we have n jobs $j_1, j_2, \ldots, j_n$, with corresponding runtimes $p_1, p_2, \ldots, p_n$. The jobs are to be scheduled on m identical machines (or processors) so as to minimize the overall finish time. Consider the *list-approx-algorithm* A, that considers the n jobs one by one, and assigns the current one to the machine that is the least loaded. The load of a machine is the total run-time of the jobs assigned to it.

a. Show that $\frac{A(I)}{A^*(I)} \leq 2 - \frac{1}{m}$, where A^* is the exact algorithm that gives the optimum solution.

b. Show that the bound is tight. In other words, find a problem instance for which $\frac{A(I)}{A^*(I)} = 2 - \frac{1}{m}$.