CSC 364H (L0101) **Test 1** **Summer 99**

No aids allowed. Answer all questions on test paper. Use backs of sheets for scratch work.

Total Marks: 30

[6] 1. *Proof of Lemma*: Assume that $\overline{M} \neq M$ and $\overline{M} \leq \frac{1}{2}C$, in order to get a contradiction. Now, let S_0 be the result of greedy, i.e. a subset of $\{1, \ldots, d\}$ such that $K(S_0) = \overline{M}$.

(a) Show that there is at least one weight in S_0:
If $S_0 = \emptyset$, then $\overline{M} = 0$, and all weights must be larger than C, but then $M = 0$, so $\overline{M} = M$ which is not the case by assumption.

(b) Show that there is at least one weight not in S_0:
If all the weights are in S, then also $\overline{M} = M$.

(c) Show now that $\overline{M} > \frac{1}{2}C$ (and hence get the contradiction) by considering the first weight, call it w_j, which has been rejected after at least one weight has been added:
How do we know that there is such a weight w_j? From (a) and (b) we know that some weights are in S_0 and some are not in S_0, but it can still happen that a first few weights have been rejected, and then every weight has been added. But in this case, these first few must be $> C$, and all the weights that have been added must be $\leq C$, and again $\overline{M} = M$, which is a contradiction.
Thus, we know there is such a weight w_j. If $w_j \leq \frac{1}{2}C$, then since $\overline{M} \leq \frac{1}{2}C$, the greedy algorithm would have added it. Thus $w_j > \frac{1}{2}C$. Since the weights have been rearranged by greedy in non-increasing order, the weights which have been added are also $> \frac{1}{2}C$. Thus $\overline{M} > \frac{1}{2}C$.

[12] 2. (a) Define an array of subproblems:
$R(j)$ = length of the longest monotone subsequence which ends in a_j. Then the answer would be $L = \max_{1 \leq j \leq n} R(j)$.

(b) Find a recurrence (no justification necessary):
let $R(1) = 1$, and for $j > 1$,

$$R(j) = \begin{cases} 1 & \text{if } a_i > a_j \text{ for all } i \ 1 \leq i < j \\ 1 + \max_{1 \leq i < j}\{R(i) | a_i \leq a_j\} & \text{otherwise} \end{cases}$$

(c) See the notes.

[12] 3. Suppose $k > 0$. Start with $A(k)$. If $A(k) = A(k-1)$, then no activity has been scheduled to end at time u_k, so we proceed to examine $A(k-1)$.

If, on the other hand, $A(k) \neq A(k-1)$, then we know that some activity has been scheduled to end at time u_k. We have to find out which one it is. We know that in this case

$$A(k) = \max_{1 \leq i \leq n} \{p_i + A(H(i)) | f_i = u_k\}$$

so we examine all activities i, $1 \leq i \leq n$, and output the (first) activity i_0 such that $A(k) = p_{i_0} + A(H(i_0))$ and $f_{i_0} = u_k$, and proceed to examine $A(H(i_0))$.

We stop when $k = 0$.