CSC 364H (L5101) **Test 1** **Summer 99**

Name ______________________ Student No. ______________________

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

Total Marks: 30

[6] 1. We are given n jobs to process. If job i is processed, it must begin at time b_i and finish at time e_i, where $b_i < e_i$. We have only one processor so only one job can be processed at a time. Since the begin-finish intervals for different jobs may overlap, it may not be possible to schedule all jobs. Our problem is to schedule as many jobs as possible.

A *schedule* consists of a list $i_1, i_2, \cdots, i_k$ of job numbers, where

$$b_{i_1} < e_{i_1} \leq b_{i_2} < e_{i_2} \leq \cdots \leq b_{i_k} < e_{i_k}$$

in order to ensure that the jobs do not overlap.

Describe, at a high level, a simple greedy algorithm which takes as input the numbers $b_1, e_1, b_2, e_2, \ldots, b_n, e_n$, and outputs an optimal schedule (i.e. one with the largest possible number of jobs). Do *not* prove that your algorithm is correct.

Hint: consider the finish times.

[12] 2. Recall the Simple Knapsack Problem:

Input: $d, w_1, \ldots, w_d, C \in \mathbb{N}$ (where C is the capacity).

Problem: Find $\max_S \{K(S) | K(S) \leq C\}$, where $S \subseteq \{1, \ldots, d\}$, and $K(S) = \sum_{i \in S} w_i$.

In our dynamic programming solution, we define a Boolean array $R(i, j)$ for $0 \leq i \leq d$ and $0 \leq j \leq C$, as follows:

$$R(i,j) = \begin{cases} T & \text{if } \exists S \subseteq \{1, \ldots, i\} \text{ s.t. } K(S) = j \\ F & \text{otherwise} \end{cases}$$

(where the convention is that if $i = 0$, then $\{1, \ldots, i\} = \emptyset$). Give the recurrence used for computing this array, and justify it *briefly*:

(a) Initialization:

(b) For $i > 0$, $R(i, j) = T$ iff

(c) Justification:

[12] 3. Recall that the Job Scheduling with Profits Problem is specified as follows:

Input: A list of jobs $(d_1, t_1, p_1), \ldots, (d_n, t_n, p_n)$, where each job consists of a duration $d_i > 0$, a deadline $t_i > 0$, and a profit $p_i > 0$. Assume that d_i and t_i are integers, and that $t_1 \leq t_2 \leq \ldots \leq t_n$.

Problem: Find $\max\{P(C) | C \text{ is a feasible schedule}\}$.

Recall that for a schedule $C(1 \ldots n)$, $C(i)$ is the start time of job i if $C(i) \geq 0$, and $C(i) = -1$ if job i is not scheduled. Also, the schedule is feasible if each scheduled job finishes by its deadline, and no two scheduled jobs overlap.

To give a dynamic programming solution, we define an array $A(i, t)$, for integer values of i and t, by

$$A(i,t) = \max\{P(C) | C \text{ is a feasible schedule in which only jobs in } \{1, \ldots, i\} \text{ are scheduled, and all scheduled jobs finish by time } t\}$$

(where if $i = 0$, then $\{1, \ldots, i\} = \emptyset$). We want to give a recurrence for computing this array. As a hint, to compute $A(i, t)$ in term of values $A(i-1, *)$, consider the two cases where either job i is scheduled as late as possible, so that it finishes at time $t_{\min} = \min\{t, t_i\}$, or job i is not scheduled.

Complete the following recurrence and justify it *briefly*:

(a) Initialization:

(b) To compute $A(i, t)$ for $i > 0$ first define $t_{\min} = \min\{t, t_i\}$. Now

$A(i, t) =$

(c) Justification: