**CSC 364** **Quiz #1** **October 11/2000**

Name ______________________ Student No. ______________________

NO AIDS ALLOWED

Answer ALL questions on test paper. Use backs of sheets for scratch work.

Total Marks: 33

**ADVICE: Do this page fast!**

[3] 1. True or False? All logarithms are base 2. NO JUSTIFICATION NECESSARY.

(a) $9n \log n + 10n + n^{1/2} = O(n)$

(b) $2^{4n \log n} = O(5^{5n \log n})$

(c) $8n^2 \log n + 5n^2 = O(n^3)$

**Solution:** All items except (a) are True.

2. Suppose that the available coin denominations are $1, p, p^2, ..., p^n$, where $p > 1$ and $n \geq 0$ are integers. Below is a greedy algorithm which solves the *make change* problem. The algorithm takes as inputs the triple $(p, n, N)$, where $p$ and $n$ indicate the coin denominations as just explained, and it returns a shortest possible list $L$ of coin denominations summing to $N$. For example,

$$make - change(3, 4, 11) = \langle 9, 1, 1 \rangle$$

since $11 = 9 + 1 + 1$ and three is the fewest number of coins summing to 11 among the denominations $\{1, 3, 9, 27, 81\}$.

**function** $make - change(p, n, N)$
$L \leftarrow \emptyset$ {Initialize $L$ to the empty list}
$s \leftarrow 0$ {$s$ is the sum of the coins in list $L$}
**while** $s < N$ **do**
    $x \leftarrow$ largest $p^i \leq p^n$ such that $s + p^i \leq N$
    $L \leftarrow L, x$
    $s \leftarrow s + x$
**return** $L$

Outline a proof that the algorithm is correct as follows:

Define what it means for a partial solution $L$ for is a loop invariant of your algorithm. Make clear exactly which cases and subcases must be considered. There is one more difficult subcase for which you do not need to give the argument.

There is an extraneous sentence in the above question. The question should read as follows: "Outline a proof that the algorithm is correct. Make clear exactly which cases and subcases must be considered. There is one more difficult subcase for which you do not need to give the argument."

A partial solution is promising if it can be extended to an optimal solution. We want to show that after each iteration of the Make-change algorithm, that the partial solution, $L$ is promising.

1 Initially, $L$ is empty, so any optimal solution extends $L$. Hence, $L$ is promising.

2 Suppose $L$ is promising and $s < N$. Then there is a list $L'$ so that $L, L'$ is an optimal solution.

2a If $x$ is in $L'$, then $L'$ can be reordered to $x, L''$, and so $L, x, L''$ is an optimal extension of $L, x$.

2b If $x$ is not in $L'$, it is more difficult to find an extension $L''$ so that $L, x, L''$ is optimal. (In this case, one can prove that there is a sublist $L_0$ of $L'$ whose coins sum exactly to $x$.)

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

**Input** A list of jobs $(d_1, t_1, p_1),..., (d_n, t_n, p_n)$, where job $i$ has duration $d_i > 0$, deadline $t_i > 0$, and profit $p_i > 0$. Assume that $d_i, t_i$ are integers.

**Output** A feasible schedule $C(1),...,C(n)$ such that the profit $P(C)$ is the maximum possible among feasible schedules. (All scheduled jobs must start at time 0 or later.)

Recall that for schedule $C(1...n)$, $C(i)$ 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 $t$, $0 \leq i \leq n, 0 \leq t \leq t_n$, by

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

You are to give a recurrence for computing the array. As a hint, to compute $A(i,t)$ in terms of values $A(i-1,*)$, consider the two cases that either job $i$ occurs in the optimal schedule or not. If so, we may as well assume that job $i$ is scheduled as late as possible; so that if finishes at time $t_{min} = \min\{t, t_i\}$.

Complete the following recurrence and **justify it briefly:**

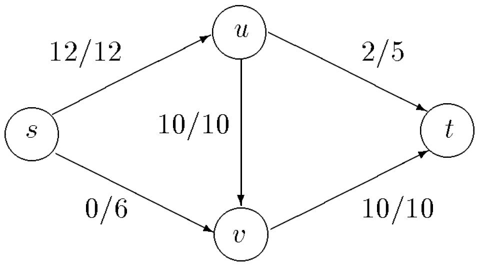
$A(0,t) = 0$, $0 \leq t \leq t_n$ (Initialization)

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

$$A(i,t) = \begin{cases} \mathsf{A(i-1,t)} \text{ if } \mathsf{t_{min} < d_i} \\ \max\{\mathsf{A(i-1,t), p_i + A(i-1, t_{min} - d_i)}\} \text{ otherwise.} \end{cases}$$

Justification:

We assume that jobs have been sorted according to deadlines so $t_1 \leq t_2 \leq \ldots \leq t_n$. Thus (as the hint suggests), if job $i$ is scheduled in the optimal schedule, it finished at time $t_{min}$, and starts at time $t_{min} - d_i$. If it is scheduled, the maximum possible profit is $A(i-1, t_{min} - d_i) + p_i$. Otherwise the maximum profit is $A(i-1,t)$.




4. In the flow network $G$ below, each flow $f$ is specified and the edge capacities are specified using the notation in the text. For example, the edge label 2/5 on the edge $(u, t)$ means that $f(u, t) = 2$ and $c(u, t) = 5$.

[2] (a) Specify a particular augmenting path $p$ for the flow $f$ by specifying a sequence of nodes below:

**Solution:** $p = \langle s, v, u, t \rangle$

[2] (b) What is the residual capacity of the path $p$ that you specified in (a)?

**Solution:** $c_f(p) = 3$

[4] (c) Specify a particular maximum flow $f^*$ and give the value $|f^*|$ below:

**Solution:**

$f^*(s, u) = 12$ $\quad$ $f^*(s, v) = 3$ $\quad$ $f^*(u, v) = 7$ $\quad$ $f^*(u, t) = 5$ $\quad$ $f^*(v, t) = 10$

$|f^*| = 15$

[4] 5. Suppose that each edge capacity of a flow network $G$ is an integer multiple of 3. Prove using the idea of cuts that the maximum flow in $G$ is an integer multiple of 3.

**Solution:** By the "Max flow min cut theorem" there is a cut $(S, T)$ such that the capacity $c(S, T)$ equals the maximum flow. But $c(S, T)$ is a multiple of 3, because it is a sum of edge capacities, each of which is a multiple of 3.