Name ______________________ Student No. ______________________

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

Total Marks: **100**

The test consists of 4 questions and 6 pages.

[25] 1. Define the problem *Dispersed Knapsack* as follows:
Input: $w_1, \ldots, w_d, C$, all positive integers, such that the w_i's satisfy the following condition:

$$w_i \geq \sum_{j=i+1}^{d} w_j, \quad \text{for } i = 1, \ldots, d-1$$

Output: $S_{\max} \subseteq \{1, \ldots, d\}$ such that $K(S_{\max}) = \max_{S \subseteq \{1,\ldots,d\}} \{K(S) | K(S) \leq C\}$.

Give a greedy algorithm which solves Dispersed Knapsack by filling in the following two blanks:

$S \longleftarrow \emptyset$
for $i : 1..d$
    if ______________________ then
        ____
    end if
end for

[25] 2. Let $G = (V, E)$ be an undirected (connected) graph where each edge $e \in E$ has a cost $c(e)$ associated with it.

(a) Describe Kruskal's algorithm for finding the minimum cost spanning tree T of G.

(b) Suppose that the costs of all the edges are distinct; that is, after ordering the edges by costs we have $c(e_1) < c(e_2) < \cdots < c(e_m)$. Show that in that case G has a *unique* spanning tree. (**Hint:** use an appropriate definition of promising.)

(*Use next page if necessary.*)

(*empty page*)

[25] 3. Consider the following variation of the Longest Monotone Subsequence problem:

Input: $d, a_1, a_2, \ldots, a_d \in \mathbb{N}$.

Output: What is the length of the longest subsequence of $a_1, a_2, \ldots, a_d$, where any two consecutive members of the subsequence differ by at most 1?

For example, the longest such subsequence of $\{7, 6, 1, 4, 7, 8, 20\}$ is $\{7, 6, 7, 8\}$, so in this case the answer would be 4.

Give a recurrence for this problem.

[25] 4. An *activity* i has a fixed start time s_i, finish time f_i, and profit p_i. Given a set of activities, we want to select a subset of non-overlapping activities with maximum total profit.

Input: A list of activities (s_1, f_1, p_1),..., (s_n, f_n, p_n). Assume $p_i > 0$, $s_i < f_i$, and $s_i, f_i, p_i \in \mathbb{R}$ where $1 \leq i \leq n$.

Output: Find a set $S \subseteq \{1, \ldots, n\}$ of selected activities such that no two selected activities overlap, and the profit $P(S) = \sum_{i \in S} p_i$ is as large as possible.

To solve this problem, we sorted the activities by their finish time: $f_1 \leq f_2 \leq \ldots \leq f_n$. Then we partition the activities according to their finish times, and denoted these *distinct* finish times by $u_1 < u_2 < \ldots < u_k$.

Let u_0 be $\min_{1 \leq i \leq n} s_i$, i.e., the earliest start time. Thus, $u_0 < u_1 < u_2 < \ldots < u_k$. We define an array $A(0..k)$ as follows:

$$A(j) = \max_{S \subseteq \{1,\ldots,n\}} \{P(S) | S \text{ is feasible and } f_i \leq u_j \text{ for each } i \in S\}.$$

where S is *feasible* if no two activities in S overlap.

Note that $A(k)$ is the maximum possible profit for all feasible schedules S.

Your job is to answer the following question:

Given that A has been computed, how do you find an actual set of activities S such that $P(S) = A(k)$?

(*Use next page if necessary.*)

(*empty page*)

End of Test