First Person:  
Family Name:  
Given Name:  
Student #:  
Email:  

Second Person:  
Family Name:  
Given Name:  
Student #:  
Email:  

Guidelines:

- You may complete this assignment alone or in groups of two. Do not get solutions from other pairs. Though you are to teach & learn from your partner, you are responsible to do and learn the work yourself. Write it up together.

- Please make your answers clear and succinct.

- Relevant Readings:
  
  - CLRS Ch. 15-16
  - Edmonds Ch. 16, 18, 19

- This page should be the cover of your assignment.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (40 marks)</td>
<td></td>
</tr>
<tr>
<td>2 (30 marks)</td>
<td></td>
</tr>
<tr>
<td>3 (30 marks)</td>
<td></td>
</tr>
<tr>
<td>Total (100 marks)</td>
<td></td>
</tr>
</tbody>
</table>
1. (40 marks) **Descending Kruskal’s algorithm**

Another approach to the minimum spanning tree algorithms is to remove “heavy” edges from the graph until only the tree is left, rather than adding “light” edges to an originally empty graph.

(a) (20 marks) Design a version of Kruskal’s algorithm that would exploit this idea. Your algorithm should work by removing heavy edges from the original graph until only an MST is left. Provide pseudo-code for your algorithm.

(b) (20 marks) Prove that your algorithm always returns an MST of $G$. The steps of your proof will be similar to that of Kruskal’s algorithm. Clearly identify your loop invariant and prove that it is established and maintained.

2. (30 marks) **SubsetSum (greedy algorithms)**

A SubsetSum is defined as follows: given positive integers $a_1 \ldots a_n$ (not necessarily distinct), and a positive integer $t$, find a subset $S$ of $(1 \ldots n)$ such that $\sum_{i \in S} a_i = t$, if it exists.

(a) (10 marks) Suppose each $a_i$ is at least twice as large as the sum of all smaller numbers $a_j$. Give a greedy algorithm to solve SubsetSum under this assumption.

(b) (20 marks) Prove correctness of your greedy algorithm by stating and proving the loop invariant.

3. (30 marks) **SubsetSum (dynamic programming)** Now suppose that the $a_i$ values are arbitrary. Design a dynamic programming algorithm to solve the SubsetSum problem. The running time of your algorithm should be polynomial in both $n$ and $t$.

(a) (10 marks) Give the definition of the array $A$ you will use to solve this problem and state how you find out if there is such a set $S$ from that array.

(b) (10 marks) Give the recurrence to compute the elements of the array $A$, including initialization.

(c) (5 marks) State how you would recover the actual set $S$ given $A$.

(d) (5 marks) Analyze the running time of your algorithm (including the step reconstructing $S$), in terms of $n$ and $t$. 