Instructions: Do not open until the exam starts. The exam will run for 45 minutes. The problems are roughly sorted in increasing order of difficulty. Answer all questions completely (though pay attention to exactly what the question is asking for). You are free to make use of any result in the textbook or proved in class. You may use up to 6 one-sided pages of notes, and may not use the textbook nor any electronic aids. Write your solutions in the space provided, the blank page after this one, or on the scratch paper provided (be sure to label it with your name). If you have solutions written anywhere other than the provided space be sure to indicate where they are to be found.

If the problem asks for an algorithm, giving a correct algorithm with worse runtime efficiency than what is asked for will be awarded partial credit.

Please sit in the seat denoted below.

Name:

ID Number:

Seat:

<table>
<thead>
<tr>
<th>Problem</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This page is left blank for scratch work.
Question 1 (Sorting Runtime, 30 points). Consider the following sorting algorithm:

RecursiveSort(A, i, j) \ \ sorts the elements of A between indices i and j
  if j=i+1
    if A[i]>A[j]
      swap(A[i],A[j])
  else
    Set k = ceiling(2*(j-i+1)/3)-1
    RecursiveSort(A, i, i+k)
    RecursiveSort(A, j-k, j)
    RecursiveSort(A, i, i+k)

(a) Give a recurrence for the runtime of this algorithm on an array of length n (when i = 1 and j = n). [10 points]

(b) What is the asymptotic runtime of this algorithm? [20 points]
Question 2 (Escape, 35 points). Ashley is trying to escape from a burning building. She has an undirected graph $G$ mapping out the building, with edge weights denoting how long it will take her to traverse each edge. Her current location is a vertex $s$ and the exit is at a vertex $t$. Unfortunately, as the building burns, some of the vertices may become inaccessible. Each vertex $v$ has a time $T_v$ after which it will become impassible.

Give an algorithm that given $G, s, t$ the weights and the times $T_v$ determines whether or not it is possible for Ashley to escape. For full credit, your algorithm should run in time $O(|V| \log(|V|) + |E|)$ or better.

For full credit, your algorithm should run in time $O(\log(n))$. 