1. [6 marks] Here is a min-heap in tree form:

```
   30
  /   \
 50   40
 / \
70 90 95
```

I now insert a priority \( R \). \( R \) is chosen randomly from \{20, 31, 67, 83\} with these probabilities:

\[
\begin{align*}
Pr(R = 20) &= 0.1 \\
Pr(R = 31) &= 0.4 \\
Pr(R = 67) &= 0.3 \\
Pr(R = 83) &= 0.2
\end{align*}
\]

Note that if 67 is chosen, then the insert performs one swapping; if 31 is chosen, then the insert performs two swapping; etc.

What is the expected value of the number of swappings? Show your steps.

2. Here is a randomized program:

```plaintext
c := 1
for i = 1 to 17:
  if random() < 1/2^c then c := c + 1
```

random() returns a real number in \([0, 1]\) randomly chosen with uniform probability. In other words, \( random() < 1/2^c \) is true with probability \( 1/2^c \), false with probability \( 1 - 1/2^c \).

Let the random variable \( X \) be the final value of \( c \). Give and justify formulas for:

(a) [4 marks] \( Pr(X = 1) \)
(b) [6 marks] \( Pr(X = 2) \)

3. Insert these string keys into a hash table, in the order shown:

- "albert"
- "bart"
- "carol"
- "daisy"
- "edward"

Hash code: sum the ASCII codes of the characters, e.g., "bart" becomes 98 + 97 + 114 + 116.

Compression map: \( k \mod 7 \).

So the array has length 7, i.e., indexes are from 0 to 6 inclusive.

Collision handling: See the subquestions.

(a) [2 marks] What are the hash codes of the keys?
(b) [2 marks] What are the array indexes obtained from the compression map?
(c) [5 marks] Starting from scratch, insert the keys using chaining for collision handling. Show the array at the end.
(d) [5 marks] Starting from scratch, insert the keys using linear probing for collision handling. Note that the order of insertions matters. Show the array at the end.

4. [5 marks] My friend John is implementing a hash table. When the array length just needs to be 30, he uses this compression map: \( 12 \cdot k \mod 30 \). What goes wrong, and why?