Due: October 15, 1992

1. (3 points) Find simple forms for \( \sum_{1 \leq k \leq n} (2^k - 1) \) and \( \sum_{0 \leq k} \frac{(k-1)}{2^k} \).

2. (2 points) Show that \( \sum_{1 \leq k} \frac{1}{k^2} \) is bounded above by a constant.

3. (5 points) Robin hood hashing is used to reduce the variance of the expected successful search time for a key in an open hash table. When two keys collide during an Insertion, the key that has the longest probe sequence stays in that position, and the other continues probing. Test this by declaring a hash table for a reasonably large value of \( m \), deriving a source of \( n = .95 \times m \) random keys, and finding a "good" hash function. Extend the hashing technique to associate with each key \( x \), the length of its probe sequence \( p(x) \).

A) Create a hash table \( T \), Insert the \( n \) keys into \( T \) using linear probing, and then use the lengths of the \( n \) probe sequences to estimate the mean and variance of \( p \). Note that (although you needn't use this) \( E[p] = Error! \) - Error! + \( O(m^{-2}) \)

B) Create a hash table \( T \), Insert the \( n \) keys into \( T \) using Robin Hood and linear probing, and then use the lengths of the \( n \) probe sequences to estimate the mean and variance of \( p \).
1. \[ \sum_{1 \leq k \leq n} (2k - 1) = \sum_{1 \leq k \leq n} 2k - \sum_{1 \leq k \leq n} 1 = 2 \sum_{1 \leq k \leq n} k - n(n+1) = n(n^2-1) \]

\[ \sum_{0 \leq k} \frac{(k-1)}{2^k} = \frac{\sum_{0 \leq k} (1/2)^k}{\sum_{0 \leq k} (1/2)^k} = \frac{1}{(1/2)^2} - 1 = 2 - 2 = 0 \]

2. Because \(1/k^2\) is a continuous decreasing function for \(k \geq 1\),

\[ \sum_{1 \leq k \leq n} \frac{1}{k^2} = 1 + \sum_{2 \leq k \leq n} \frac{1}{k^2} \leq 1 + \int_1^n \frac{1}{k^2} = 1 + \frac{1}{-\frac{1}{n}} = 1 + \left( -\frac{1}{n} + 1 \right) \]

\[ \sum_{1 \leq k \leq n} \frac{1}{k^2} \approx 2 - \frac{1}{n} < 2 \]

3. For Think C on a MacSE, \(m = 997, n = 950, h(x) = x \bmod m\)

For Linear Probing: \(E[p] = 5.21, V(p) = 141.21\)

For Robin Hood with Linear Probing: \(E[p] = 5.21, V(p) = 9.98\)

For \(m = 997\) and \(n = 950\), \(E[p] = 6.319346387601973\)

typedef int key_type;

typedef struct {
    key_type key;
    int probe_length;
} element;

typedef element table[m];

void create_table(T)
{
    int k;
    for (k=0;k<m;k++)
    {
        T[k].key=EMPTY;
        T[k].probe_length=0;
    }
}
void Insert(T, x)
table T;
key_type x;
{
    int i, b, p_length = 0;
    b = h(x);
    while(1){
        p_length++;
        if (T[b].key == EMPTY) {
            T[b].key = x;
            T[b].probe_length = p_length;
            return;
        }
        else if (T[b].key == x) return;
        else b = (++b) % m;
    }
}

void RobinHoodInsert(T, x)
table T;
key_type x;
{
    int i, b, p_length = 0;
    b = h(x);
    while(1){
        p_length++;
        if (T[b].key == EMPTY) {
            T[b].key = x;
            T[b].probe_length = p_length;
            return;
        }
        else if (T[b].key == x) return;
        else if (T[b].probe_length >= p_length) b = (++b) % m;
        else { swap(T, b, &x, &p_length); b = (++b) %