2) (10 pts) ANL (Summations and Algorithm Analysis)

Give the big-oh runtimes for each of the following functions in terms of *n* and/or *k* (where *k* is the length of string *s*), given that *strlen*(*s*) is an O(*k*) function and *toupper*(*c*) is an O(1) function. You may assume that *s* is non-NULL and contains at least one character (so, it shouldn't cause any of the following functions to crash).

```
void uppercase(char *s)
{
   int i;
   for (i = 0; i < strlen(s); i++)
      s[i] = toupper(i);
}
```

uppercase run time: $\boldsymbol{O(k^2)}$

```
void uppercase_remix(char *s)
{
   int i, length = strlen(s);
   for (i = 0; i < length; i++)
      s[i] = toupper(i);
}
```

uppercase_remix run time: $\boldsymbol{O(k)}$

```
void uppercase_unreliable(char *s){
   int i, j = strlen(s) - 1, m;
   while (i <= j)
   {
      m = i + (j - i) / 2;
      if (rand() % 2 == 0)
      {
         s[i] = toupper(s[i]);
         i = m + 1;
      }
      else
      {
         s[j] = toupper(s[j]);
         j = m - 1;
      }
   }
}
```

uppercase_unreliable run time: $\boldsymbol{O(k) + O(\log k) = O(k)}$

```
void mad_scramble(char *s, int n){
   int i;
   for (i = 0; i < n; i++)
      s[strlen(s) - 1] = rand() % 25 + 'a';
}
```

mad_scramble run time: $\boldsymbol{O(nk)}$

Grading: 3 pts uppercase, 2 pts uppercase_remix, 3 pts uppercase_unreliable, 2 pts mad_scramble, each part is all or nothing.