Com S 311 Programming Project 1 John Collin Vincent

WarWithArray:

```
List 2k;
quicksort(s);

for each i in s
  for each j in s
    if(i != j)
      2k.add(i+j);

for each i in 2k
  for j = k+1 to i.length()
    temp = i.substring(j-k, j);
    if(!s.binarySearch(temp))
      2k.remove(i);
return 2k;
```

this starts with quick sort which has a time complexity of $n \log n$, it then constructs the list of all possible 2k length strings which takes n^2 time. Then it loops through the list of 2k strings which is $n(n-1)$ long, and for each of those it does $k-1$ binary searchs which takes $k(\log n)$ time since it takes k time to compare the strings. So the total time complexity is $n \log n + n^2 + n(n-1)(k-1)(k \log n)$

WarWithBST:

```
List 2k;
BST bs;

for each i in s
  bs.add(i);
  for each j in s
    if(i != j)
      2k.add(i+j);

for each i in 2k
  for j = k+1 to i.length()
    temp = i.substring(j-k, j);
    if(!bs.search(temp))
      2k.remove(i);
return 2k;
```

this start with inserting the k length strings into a bst, the insertion takes up to $\log n$ time and there are n strings to insert, it also must create the list of $2k$ strings which takes n^2 time. then like before it loops through the $n(n-1)$ 2k strings and does up to $k-1$ searches through the BST which each take $k\log n$ time. The total time complexity ends up being $n\log n+n^2+n(n-1)(k-1)(k\log n)$

WarWithHash: