Assignment #2

Casey Boardman
cdb@ccs.neu.edu
CSG 399

February 23, 2004

1 Problem 1

In the longest substring problem, we are given a set of string $S = s_1, \ldots, s_n$ and are asked to determine the longest string s that is a substring of all the n strings. Is the longest substring solvable in polynomila time? Provide an answer by either giving a polynomial-time alogorithm or proving that the problem is NP-complete.

Here is a polynomial-time solution.

Without loss of generality, assume that the stings are in order by length, shortest to longest. s_1 is therefore the shortest string. The longest common substring can be at most as long as s_1 since it must be a substring of s_1.

```
for each substring, s, of s_1
   for each s_i, i=2 to n
      if s is a s_i
         good!  loop.
      else
         s = longest substring of s, s_i
```

Note that we have to try every substring of s_1 (the outer loop) since if we did not we may not find the optimal string. Example: The longest substring of s_1 and s_2 is the first 10 characters of s_1. We will then limit our searching to the first 10 characters of s_1. It may work out that the longest common substring is the last 5 characters of s_1 (assume a 15 character s_1 for this example). By trying all substring of s_1 we avoid this issue.

Notice that this algorithm is polynomial time given that we can find the longest substring of two strings in polynomial time. This is possible. There is, in fact, an algorithm for the longest common *subsequence* (a harder problem) in CLRS pg 350-356 in $O(mn)$ time.