

Homework 9 Write-up

Luke Levis

November 11, 2014

Main Idea: I came up with the greedy algorithm for solving the shortest substring problem. I start with the set S of the reads, and I remove substrings within the list. Then, while the size of the set is larger than one, I create all possible pairs of the reads, calculate the pair with maximum overlap, and merge that pair together, removing the original two strings from the set, and replacing it with the merged read. At each iteration, I also remove substrings to rid the set of redundancy that could cause bugs.

Pseudocode:

```
function GETSUPERSTRING(reads)
  removeSubs(reads)
  while  $\text{len}(\text{reads}) > 1$  do
    pairs := makePairs(reads)
    max := maxOverlap(pairs)
    merged := merge(max)
    reads.remove(max[0])
    reads.remove(max[1])
    reads.append(merged)
    removeSubs(reads)
  end while
  return reads[0]
end function
```

Runtime Analysis: The running time of my algorithm is:

$$\Theta(n^3k)$$

There is $O(k)$ time to compute the overlap between two strings, via a linear scan that is done in a separate method I wrote (see python code). The time to create pairs of reads is $O(n^2)$, and for each loop we remove the original strings from the list which takes $O(n)$ time. All in all, the running time is $\Theta(n^3k)$, since the time to find the max overlap is just a linear scan.