

ASSIGNMENT 1

BIOINFORMATICS (CIS 455)

Javier Arechalde

February 5, 2018

Problem 1-1 Jones & Pevzner, Problem 2.1

1.

We are going to write the pseudocode for an algorithm, that given a list of n numbers, returns the largest and smallest number in that list.

Algorithm 1 Finding largest and smallest numbers

```
1: Initialize  $max$  to  $L[0]$  and  $min$  to  $L[0]$ 
2: for  $n$  in number list  $L$  do
3:   if  $n < min$  then
4:      $min = n$ 
5:   end if
6:   if  $n > max$  then
7:      $max = n$ 
8:   end if
9: end for
10: Return  $max$  and  $min$ 
```

The time complexity of my algorithm will be $O(n) = n$, and the running time will be $2(n - 1) + 2$ because for every number in the list that starts from the second position after initialization, we need to check if the number is greater than the current maximum, or smaller than the current minimum.

Now we will implement an algorithm that performs only $3n/2$ comparisons to find the smallest and largest numbers in the list. For this implementation we will use the tournament method.

Problem 1-2 Jones & Pevzner, Problem 2.2

Now we will write the pseudocode for two algorithms that iterate over every index from $(0, 0, \dots, 0)$ to (n_1, n_2, \dots, n_d) , one of them will be recursive and the other one iterative.

Iterative pseudocode:

Algorithm 2 Iterative pseudocode

```
1: We have two indexes lists  $l = (0, 0, \dots, 0)$  and  $n = (n_1, n_2, \dots, n_d)$ .
2: We will print all the possible combination of elements
3: for Element  $l_i$  in list  $l$  do
4:   for Element  $n_i$  in list  $n$  do
5:     Print  $l_i, n_i$ .
6:   end for
7: end for
```

Recursive pseudocode:

Algorithm 3 Recursive pseudocode

```
1: We have two indexes lists  $l = (0, 0, \dots, 0)$  and  $n = (n_1, n_2, \dots, n_d)$ .
2: We will calculate the sum of the logarithm with the base of the numbers of
   the second list, using the first list of numbers as the base of the logarithm.
3: We initialize:  $sum = 0$ 
4: for Element  $l_i$  in list  $l$  and element  $n_i$  in list  $n$  do
5:    $sum = sum + \log_{l_i} n_i$ 
6: end for return The  $sum$  value
```

Problem 1-3 Jones & Pevzner, Problem 2.3

- Yes, $\log n = O(n)$, because $O(n)$ is an upper bound for $\log n$, as it grows faster.
- No, $\log n = \Omega(n)$, because $\Omega(n)$ is not a lower bound for $\log n$, as $\log n$ grows significantly slower.
- No, $\log n = \Theta(n)$, because $\log n$ is not $O(n)$ and $\Omega(n)$ at the same time.

Problem 1-4 Jones & Pevzner, Problem 2.17

Will the viruses eventually kill all the bacteria?

In order to find if the viruses end up killing all the bacteria in the Petri dish, we only need to prove if the growth of the viruses is higher than the growth of the bacteria.

In the first minute, the virus kills one bacteria, and produces another copy of himself, and all the remaining bacteria reproduce, making 2 viruses and $2(n-1)$ bacteria.

Then the viruses number will continue to double each step, and the number of bacteria will be the number of initial bacteria minus the bacteria that is killed by every virus at each step, and then doubled. The key here is that the virus continue to double at each step, and even though the number of bacteria doubles at each step too, this number is reduced by the number of viruses, before doubling, so we can say that the rate of growth of the viruses is higher than the bacteria growth rate, and at some point they will be exterminated by the viruses.

Algorithm design

Algorithm 4 Algorithm for calculating the number of steps

- 1: At the beginning we have $n_v = 1$ virus and $n_b = n$ bacteria
 - 2: At each step, the number of viruses double, and the bacteria double too, but n_v are killed by the virus too before they double.
 - 3: **while** $n_b > 0$ **do**
 - 4: $n_b = 2 * (n_b - n_v)$
 - 5: $n_v = 2 * n_v$
 - 6: **end while**
-