BIO2350 Basics of Programming and Algorithms for Bioinformatics
Exam 16.12.2015 / Juha Kesseli

1) Explain the following computer science terms as clearly as you can, using 1-4 sentences for each:
   a) Public key cryptography (1p)
   b) Pipes in UNIX (1p)
   c) Hash function (1p)
   d) Garbage collection (1p)
   e) Merge conflict (1p)
   f) Unicode (1p)

2) Number systems:
   a) Let 00111001 be an 8-bit unsigned integer in binary format (the most significant bit is on the left). Write its value in decimal format. (1p)
   b) Write the same value in hexadecimal format (1p)
   c) x=127 is an 8-bit signed integer. If you calculate x+1 on a computer, what happens? Why? (2p)
   d) If you calculate 0.1+0.2 in Python and get 0.30000000000000004 as an answer, what happened? Why? (2p)

3) Algorithms and data structures:
   a) Let's say you have a sorted table of N patient id:s in a hospital database, and you decide to use the binary search algorithm to find a patient. What is the worst case run time of this search in O-notation? (1p) How many extra steps does the binary search algorithm have to perform in the future if the number of patients in the hospital system doubles? (1p)
   b) The following Python script reads a tab-separated file containing on each line e.g. the gene name (4th column), the size of the gene (5th column) and expression levels of the gene from different samples (columns after that). Explain what the data structures created are like, what they contain and how they are created in the code. Explain the purpose, function, data structures and logic of each line in the code (4p)

```
class Gene:
  def __init__(self,name,size,expression):
    self.name=name
    self.size=size
    self.expression=expression
file=open('pc_gene_expression.tsv')
header=next(file).rstrip().split('\t')
patients=[bamname[:-4] for bamname in header[5:]]
genes=[]
for line in file:
  data=line.rstrip().split('\t')
  print(data)
  genes.append(Gene(data[3],int(data[4]),[float(num) for num in data[5:]]))
file.close()
```

4) Programming techniques:
   a) What are the differences between interpreted and compiled programming languages? What are the main benefits and drawbacks of each? (2p)
   b) The following Python script shows a programming approach to calculate arbitrary numbers from what is called the Fibonacci sequence. What is this approach called? (1p) In what main alternative way could you calculate the same output for any n? What would that solution look like (either code in Python or just all operations explained precisely)? (2p) Which approach is more computationally efficient when calculating e.g. fib(100)? Why? (1p)

```
def fib(n):
  if n == 0:
    return 0
  elif n == 1:
    return 1
  else:
    return fib(n-1) + fib(n-2)
```

5) How would you apply the nearest neighbor classification algorithm to the problem of facial recognition? What are the steps needed? How does the algorithm work? How would you test how well your classifier works? (6p)