COMP 314 Homework Assignment F

This is version 3, published 3/11/15. Version 1 included only problems for Chapter 10. Version 2 included problems for Chapter 11 too, and made minor changes to Chapter 10 questions also. Version 3 completely replaces question F5(a), and made further minor changes.

Chapter 10

Question F1. (12 points) Give the dominant term of each of the following functions.

(a) $f(n) = (\log_{10}(7n + 2))^3 + 9(\log_2(n^7))^2$

(b) $f(n) = 5n^7 + 2^{3n}$

(c) For some constants $a, b > 0$: $g(n) = an^3 \log_2(n) + b(n \log_2(n))^3$

(d) $f(n) = g(h(n))^2$, where $h(n) = n^2 \log_2 n$ and $g(n) = 3n^2 + 5n^4 + 6$

Question F2. (10 points) Given the following collection of functions f_1, f_2, f_3, f_4, list all pairs of the functions such that $f_i \in O(f_j), i \neq j$.

$$f_1(n) = n(\log n)^8$$
$$f_2(n) = 5n^2 + 4(n + 6)^3 2^n$$
$$f_3(n) = n^3 2^n$$
$$f_4(n) = f_1(n) f_3(n)$$

Question F3. (Ungraded) Answer true or false to the following:

(a) $5n^3 \in O(n^3 \log n)$

(b) $n^2 n! \in O(2^n)$

(c) $\log(5n^3 + n \log n) \in O(\log n)$

(d) $\log n \in O(\log(5n^3 + n \log n))$

(e) $(6n^2 - 2n - 4)^3 \in O(n^7)$

(f) n^8 is sub-exponential

```
# Input is a nonnegative integer M in decimal notation. Output is M^10.
# We could compute this efficiently using the Python ** operator,
# but here we deliberately use a slow iterative method.
def slow10thPower(inString):
    M = int(inString)
    product = 1
    for i in range(10):
        product = product * M
    return str(product)
```

Figure 1: The Python program `slow10thPower.py`.

```
# The input should be a positive integer M in decimal notation. This
# program returns a list of all positive even integers less than M,
# separated by commas.
def listEvens(inString):
    M = int(inString)
    evens = [ ]
    for i in range(2,M):
        if i % 2 == 0:
            evens.append(str(i))
    return ','.join(evens)
```

Figure 2: The Python program `listEvens.py`.

(g) n^8 is super-polynomial

(h) $n!$ is sub-exponential

(i) $n!$ is super-polynomial

Question F4. (Ungraded) What is the exact running time of the `containsGAGA` Turing machine (Figure 10.1, page 211) on input:

(a) `GGGGGAGAGGGGG`

(b) `GGGGGTGTGGGGG`

Question F5. (25 points) Make a reasonable estimate of the time complexity of the following Python programs:

(a) `slow10thPower.py` (Figure 1 above)

```
# This is a strange and useless program, but it is good for practicing
# the skill of estimating time complexity. The input is assumed to
# be a positive integer M in decimal notation.
def mysteryMultiply(inString):
    # make M copies of the input concatenated together
    copiedInString = int(inString) * inString
    # convert to integer and perform the mystery multiply
    val = int(copiedInString)
    return str(val*val*val)
```

Figure 3: The Python program `mysteryMultiply.py`.

(b) `countLines.py` (Figure 3.1, page 44)

(c) `pythonSort.py` (Figure 4.2, page 61)

(d) `containsMarsupial.py` (Figure 8.2, page 160)

(e) `listEvens.py` (Figure 2 above)

(f) (Ungraded) `mysteryMultiply.py` (Figure 3 above)

Question F6. (10 points) Suppose `foo.py` is a Python program with time complexity $O(n^4)$. Give a reasonable estimate of the time complexity of simulating `foo.py` on each of the following computational models (or, if it is not possible to give a reasonable estimate, explain why):

(a) a standard Turing machine

(b) a multi-tape Turing machine (with multiple independent heads)

(c) a random access Turing machine

(d) a multi-tape universal Turing machine (with multiple independent heads)

(e) a quantum computer

Question F7. (18 points) This problem concerns only the following complexity classes: Const, Lin, LogLin, Quad. For each of the computational problems given below, state which of the above complexity classes the problem belongs to. Notes: (i) each problem accepts a single ASCII string as input; (ii) a problem can belong to multiple complexity classes.

(a) CountCs: Solution is the number of `C`'s in the input.

(b) CHAR10000: Solution is the 10,000th character of the input string, or "`no`" if the input is shorter than 10,000 characters.

(c) PAIRSOFSTWORDS: Input is split into words w_i separated by whitespace. Solution is a list of all ordered pairs (w_i, w_j) of these words such that w_i starts with `S` and w_j starts with `T`. Each ordered pair appears on a separate line with the two elements separated by a space character.

(d) ISSORTED: Input is split into words separated by whitespace. Solution is "`yes`" if the words are sorted in shortlex order, and "`no`" otherwise.

(e) NUMERICALSORT: Input is split into words separated by whitespace. Words that do not represent positive integers in decimal notation are ignored. Solution is a list of the words that do represent positive integers, sorted in increasing numerical order and separated by space characters.

(f) ENDSINZ: Solution is "`yes`" if the input ends in `Z`, and "`no`" otherwise.

Chapter 11

Question F8. (Ungraded) A *polylogarithmic function* is a function of the form $p(\log n)$, for some polynomial p. For example, $6(\log n)^3 + 3\log n + 7$ is a polylogarithmic function. Define the complexity class PolyLogTime to be the set of computational problems that have a polylogarithmic time solution (i.e. there is a $O(p(\log n))$ solution for some polynomial p).

(a) Is PolyLogTime a subset of Poly? Give a brief proof of your answer.

(b) Is PolyLogTime a *strict* subset of Poly? Give a brief proof of your answer.

Question F9. (Ungraded) Define the complexity class PolyPolyLog to be the set of computational problems that have a solution in $O(p(n)q(\log n))$, for some polynomials p and q. Prove that Poly and PolyPolyLog are the same complexity class.

Question F10. (8 points) Consider the input string I = "`1 2 3 4 5 6 7 8 9 10`". How many sets would be in the solution of the following problem instances?

(a) ALL3SETS(I)

(b) ALLSUBSETS(I)

Question F11. (12 points) Give a solution to each of the following problem instances.

(a) TSP("`a,b,3 b,c,2 c,d,1 d,e,2 e,a,1 b,d,5`")

(b) TSPPath("`a,b,3 b,c,2 c,d,1 d,e,2 e,a,1 b,d,5 ; a c`")

(c) ShortestPath("`a,b,3 b,c,2 c,d,1 d,e,2 e,a,1 b,d,5 ; a c`")

(d) TSPPath("`a,b,3 b,c,2 c,d,1 d,e,2 e,a,1 b,d,5 ; a d`")

Question F12. (21 points) For each of the following problems, describe our state of knowledge about whether the problem is in Poly and/or Exp. For example, the answer for the problem Factor is "Factor $\in$ Exp, and Factor is widely believed but not proved to be outside Poly".

(a) FactorInRange: see Figure 11.7, page 243.

(b) IsComposite: see page 243

(c) FactorUnderOneMillion (FUOM): Input is an integer M. Solution is a nontrivial factor of M that is less than 1 million, or "`no`" if no such factor exists.

(d) FactorLessThan1Percent (F1Pct): Input is an integer M, solution is a nontrivial factor of M that is less than $M/100$ or "`no`" if no such factor exists.

(e) HaltsInCubicTime: Input is a program P and input I. Solution is "`yes`" if P halts within n^3 steps on input I (where $n = |I|$), and "`no`" otherwise.

(f) HaltsIn10toNTime: Input is a program P and input I. Solution is "`yes`" if P halts within 10^n steps on input I (where $n = |I|$), and "`no`" otherwise.

(g) Power: Input is two integers M_1, M_2. Solution is $M_1^{M_2}$.

Total points on this assignment: 116