A9: FIRST ALGORITHMS

I'm not counting any chickens. —Jeff Bridges

Course: CS 5002
Fall 2018
Due: 18 Nov 2018, Midnight

OBJECTIVES

After you complete this assignment, you will be comfortable with:

- Order of functions
- Asymptotic analysis
- Calculating runtime
- Becoming familiar with binary search, selection sort
- Representing algorithms (pseudocode, english, code)

RELEVANT READING

Rosen:

- Algorithms Unlocked:
  - Chapter 1: What are Algorithms and Why Should You Care?
  - Chapter 2: How to Describe and Evaluate Algorithms
- Rosen:
  - Chapter 3.2: Growth of Functions

NEXT WEEK'S READING

- Cormen: Algorithms Unlocked, pp 40-59 (mergesort, quicksort)
- Rosen: Chapter 8.3: Divide-and-Conquer Algorithms and Recurrence Relations

EXERCISES

Question 1

For the input given below, how many times is binary search called to find the word *wordmaker*? Indicate which elements are looked at in order. (for example, does the algorithm look at element 1, then 2, then 3, ...? (5)

1. aaerially
2. aardvarks
3. aaronical
4. abashment
5. curtained
6. escocheon
7. escondido
8. flagstaff
9. guildship
10. hammerkop
11. hearsiest
12. interplay
13. jumillite
14. korfballs
15. lapboards
16. lazulitic
17. mendelson
18. mudcapped
19. nailproof
20. nailsmith
21. overwhelm
22. putrefies
23. ragouting
24. rejoining
25. sensitive
26. shelfmate
27. skellying
28. triturate
29. triturium
30. tulipwood
31. unsoberly
32. unsolders
33. wordmaker
34. wowserish
35. wrastling
36. yickering
37. yikkering
38. yobberies
39. yobbishly
40. yockernut
41. yohimbine
42. yoicksing
43. youthhood
44. youthiest
45. yumminess
46. zachariah
47. zaglossus
48. zambezian
49. zastrugas
50. zoetropic
51. zolaesque

Points: ____________ out of 5

Question 2

Rank these functions in terms of order of growth. That is, find an arrangement of functions $g_1, g_2, \ldots g_n$ such that $g_1 = \Omega(g_2), g_2 = \Omega(g_3), \ldots$. When multiple functions have the same order, please indicate so. (14)

1. $\lg(\lg n)$
2. n^2
3. $n!$
4. $(\lg n)!$
5. $(\frac{3}{2})^n$
6. $2^{\lg n}$
7. $(n+1)!$
8. $\ln n$
9. $n \lg n$
10. n^3
11. $\lg(n!)$
12. $\ln n$
13. 2^{2^n}
14. $n^2 + \log n$

Question 3

For each function $f(n)$ below, indicate whether the function $f(n)$ is $O(g(n))$, $\Omega(g(n))$, or $\Theta(g(n))$, where $g(n) = n^2$.

(a) $3x + 42$ — (a) ________ (1)

(b) $(x^2)/(x+1)$ — (b) ________ (1)

(c) $(x^3 + 2x)/(2x+1)$ — (c) ________ (1)

(d) $2x^3 + x^2 \log x$ — (d) ________ (1)

(e) $2x^2 + x^3 \log x$ — (e) ________ (1)

(f) $(x^3 + 5 \log x)/(x^4 + 1)$ — (f) ________ (1)

Question 4

For this pseudocode, describe the algorithm in English.

```
COMPUTE($n$)
$p = 0$
$q = 1$
for $i = 1$ to $n$
    $q = q \cdot i$
    $p = p + n$
return $p + q$
```

(a) Describe what the algorithm is doing in “natural language”. (5)

(b) What is the runtime of this algorithm? Use Big-Θ notation. (5)

Question 5

This question pertains to the following function.

```
MYSTERY($n$)
$r = 0$
for $i = 1$ to $n - 1$
    for $j = i + 1$ to $n$
        for $k = 1$ to $j$
            $r = r + 1$
return r
```

(a) What value is returned by this function? Express your answer as a function of n. (5)

(b) What is the worst-case running time, using Big-O? (3)

Question 6

This question pertains to the following algorithm:

Input: Array A.

```
DoSomething($A$)
$p$ = TRUE
while p
    for $i = 1$ to $A.length$
        if $A[i] > A[i+1]$
            $a = A[i]$
            $A[i] = A[i+1]$
            $A[i+1] = a$
            $p$ = TRUE
    if not p
        break
    $p$ = FALSE
    for $i = A.length - 2$ to $0$
        if $A[i] > A[i+1]$
            $q = A[i+1]$
            $A[i+1] = A[i]$
            $A[i] = q$
            $p$ = TRUE
```

(a) Describe the algorithm in English. Relate it back to an algorithm we've either talked about in class or had in our reading. (5)

(b) What is the best case run time for this algorithm in terms of Big-Θ? Provide an example of the best case input. (4)

(c) What is the worst case run time for this algorithm in terms of Big-Θ? Provide an example of the worst case input. (4)

(d) What is the average case run time for this algorithm in terms of Big-Θ? (3)

Question 7

Write pseudocode to implement the algorithm described below:

Input: an array

Output: a sorted array

For every element in an array, if the previous element is less than the current element, move to the next element. Otherwise, swap the elements and move to the previous element. If there is no previous element, just move to the next element. If there is no next element, you're done.

(a) Write pseudocode to express the algorithm. (10)

(b) What's the average case running time of this algorithm? (3)

(c) What's the worst case running time of this algorithm? Provide an example of a worst-case input. (3)

(d) What's the best case running time of this algorithm? Provide an example of a best-case input. (3)

Question 8

Suppose the following algorithm is used to evaluate the polynomial $p(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_1 x + a_0$:

Input: Array A such element $A[i] = a_i$

```
EVALUATE($A$)
$p = a_0$
$x\_pow = 1$
for $i = 1$ to $n$
    $x\_pow = x \cdot x\_pow$
    $p = p + A[i] \cdot x\_pow$
```

(a) Describe the worst-case input for this algorithm. (3)

(b) How many multiplications are done in the worst-case? How many additions? (3)

(c) How many multiplications are done on the average? (3)

(d) Can you improve this algorithm? If so, explain how and provide some pseudocode. (3)

Question 9

This question pertains to the following algorithm:

```
DOTHIS($A, i, j$)
if $i > j$
    return
$m = \lfloor (i + j)/2 \rfloor$
DOTHIS($A, i, m$)
DOTHIS($A, m + 1, j$)
if $A[j] < A[m]$
    $a = A[j]$
    $A[j] = A[m]$
    $A[m] = a$
DOTHIS($A, i, j - 1$)
```

(a) Describe what this algorithm is doing; and specifically, how it is accomplishing it. (12)

(b) Provide an example of an input that will produce the worst-case runtime. (8)

SUBMISSION DETAILS

Things to submit:

- Submit the following on Blackboard for Assignment 9:
  - The written parts of this assignment as a .pdf named "CS5002_[lastname]_A9.pdf". For example, Ben Bitdiddle's file would be named "CS5002_Bitdiddle_A9.pdf". (There should be no brackets around your name).
  - Make sure your name is in the document as well (e.g., written on the top of the first page).