**“Written” Homework #2**

Functional Analysis and Textbook Problems

(105 points)

Name: [Insert Name Here]

Date: [Insert Date Here]

Assignment: *CSc345\_HW2*

1. (10 pts) The function is defined by:

if *n* is odd

if *n* is even

Which of these sets (there could be more than one) is in? ? Prove your answer.

1. (5 pts) Are all functions in for some function ? Why?

(Hint: You might want to think about problem 3 before you answer this.)

1. (10 pts) Let and be functions . We will say is related to if . Show that this is an equivalence relation. You must prove any properties of Θ that you use.
2. (5 pts) Arrange the following expressions by growth rate from slowest to fastest.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

1. (10 pts) Hardware vendor XYZ Corp. claims that their latest computer will run 100times faster than that of their competitor, Prunes, Inc. If the Prunes, Inc.computer can execute a program on input of size *n* in one hour, what size input can XYZ’s computer execute in one hour for each algorithm with the following growth rate equations?

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

1. (10 pts) For each of the following pairs of functions, either is in , is in , or . For each pair, determine which relationship is correct. Justify your answer, using the method of limits discussed in Section 3.4.5 (*on page 68 of the Schaffer Textbook*).
2. (15 pts) Determine Θ for the following code fragments in the average case. Assume that all variables are of type **int**.
3. sum=0;

for (i=0; i<n\*n; i++)

sum++;

1. sum = 0;

for (i=1; i<=n; i++)

for (j=1; j<=n; j\*=2)

sum++;

1. sum = 0;

for (i=1; i<=n; i\*=2)

for (j=1; j<=n; j++)

sum++;

1. Assume array **A** contains a *n* values, random takes constant time, and sort takes steps.

sum = 0;

for (i=0; i<n; i++){

for (j=0; **j**<n; j++)

**A**[j]=DSutil.random(n);

sort(**A**);

}

1. Assume array **A** contains a random permutation of the values from 0 to *n* - 1.

sum = 0;

for (i=0; i<n; i++)

for (j=0; A[j]!=i; j++)

sum++;

1. (10 pts) Given an array storing integers ordered by value, modify the binary search routine to return the position of the integer with the greatest value less than *K* when *K* itself does not appear in the array. Return **ERROR** if the least value in the array is greater than *K* (*Binary Search Routine is on page 73 of the Shaffer Textbook*).
2. (10 pts) Prove that if an algorithm is in the average case, then it is is the worst case.
3. (10 pts) Use the space equation of Section 4.1.3 (*Space Equation is on page 73 of the Shaffer Textbook*) to determine the break-even point for an array-based list and linked list implementation for lists when the sizes for the data field, a pointer, and the array-based list’s array are as specified. State when the linked list needs less space than the array.
4. The data field is one byte, a pointer is four bytes, and the array holds thirty elements.
5. The data field is 32 bytes, a pointer is 4 bytes, and the array holds forty elements.
6. (10 pts) Let *Q* be a non-empty queue, and let *S* be an empty stack. Using only the stack and queue ADT functions and a single element variable *X*, write an algorithm to reverse the order of the elements in *Q*. (*The Stack ADT functions are located on page 118 and the Queue ADT functions are located on page 125 of the Shaffer Textbook*).