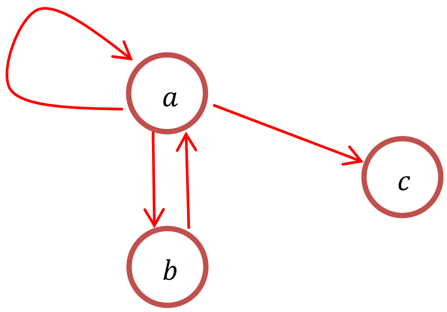
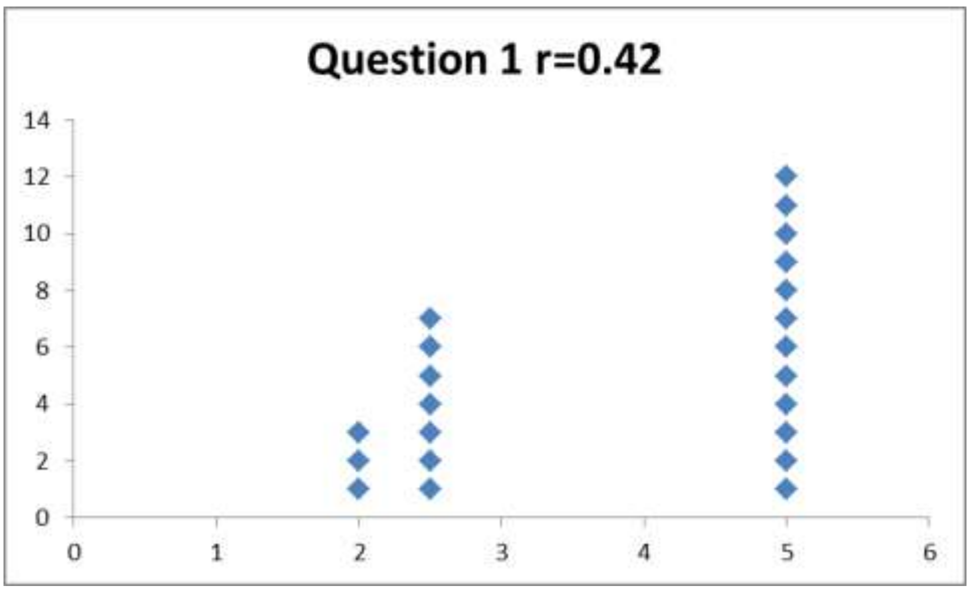
Name ____Solutions____ Discrete I, Test 3, 12/1/2014

1) Draw the digraph of the relation $\{(a,a),(a,b),(a,c),(b,a)\}$ (5 points)

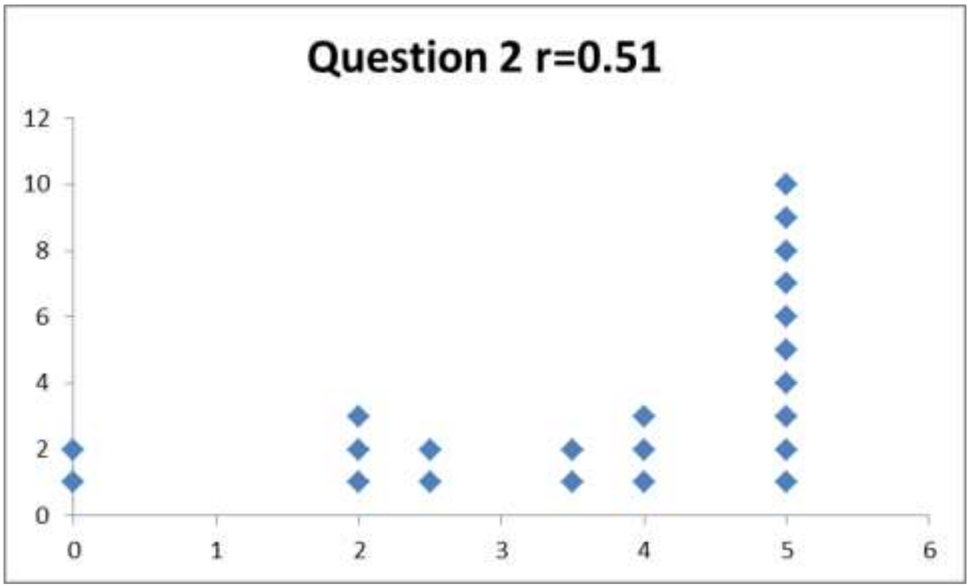




For the rest of the problems on this page, use the relation R on $\mathbb{R}$ defined by $(x, y) \in R$ if and only if $x - y = 2$

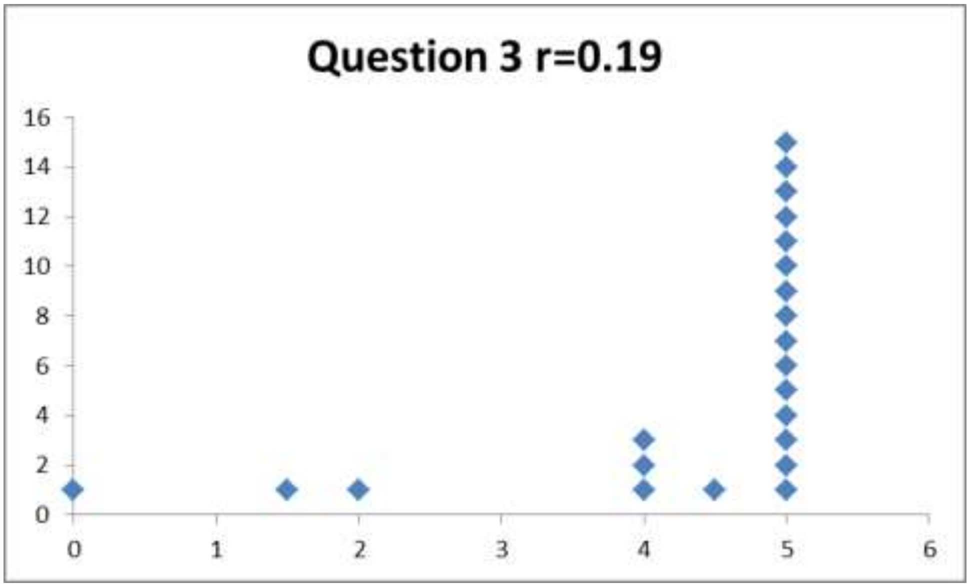
2) Is R reflexive? Justify your answer. (5 points)

No, consider $x = y = 1$. Then $x - y = 1 - 1 = 0 \neq 2$.



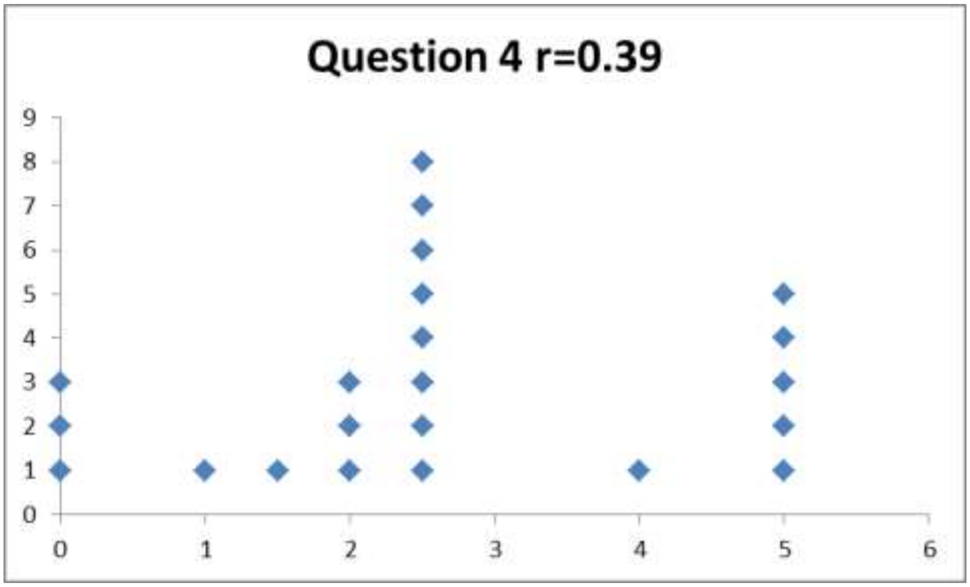
3) Is R symmetric? Justify your answer. (5 points)

No. Consider $x = 4, y = 2$. Then $x - y = 4 - 2 = 2$, so $4R2$.
On the other hand, $y - x = 2 - 4 = -2 \neq 2$.



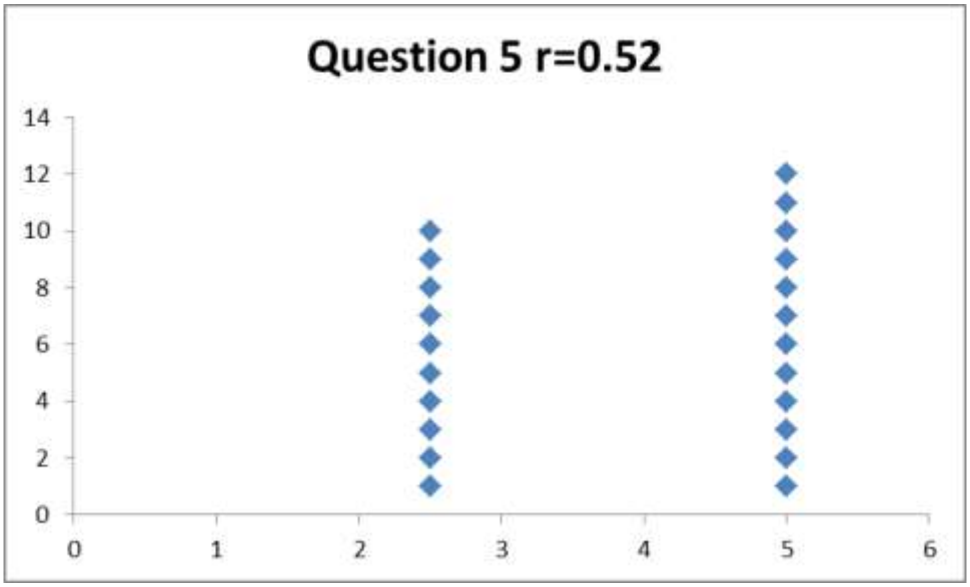
4) Is R transitive? Justify your answer. (5 points)

No. Consider $x = 4, y = 2, z = 0$. Then xRy because $4 - 2 = 2$ and yRz because $2 - 0 = 2$. However, xRz fails because $4 - 0 = 4 \neq 2$.



5) Find a partition of $\{a, b, c, d, e, f, g\}$ with 3 parts. (5 points)

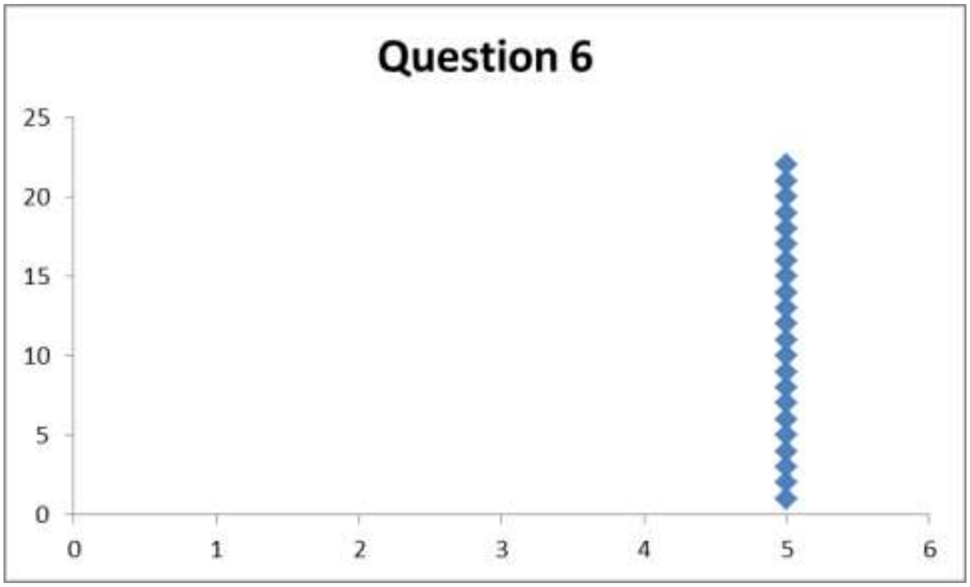
$$\{\{a, b, c\}, \{d, e\}, \{f, g\}\}$$



Compute each of the following mod 13.

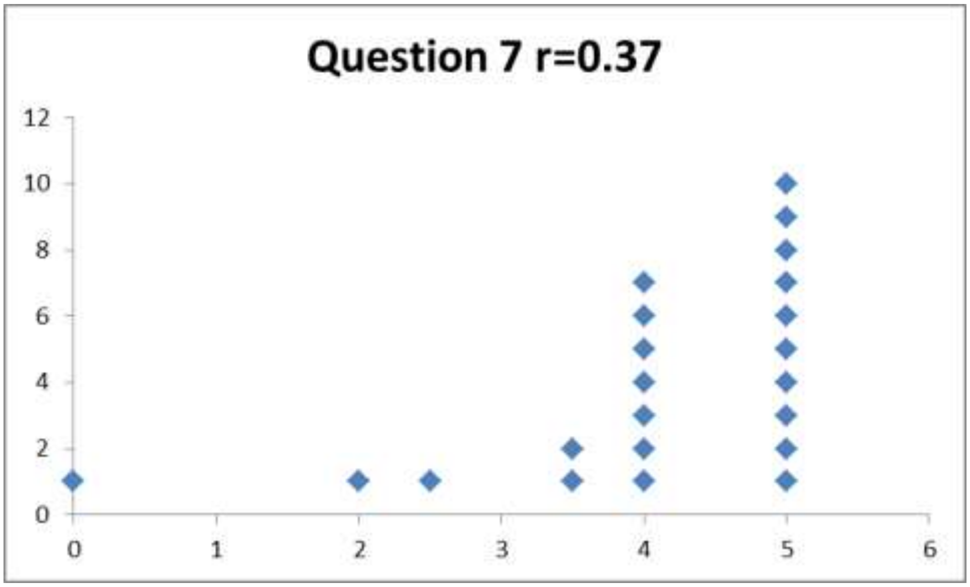
6) $10 + 8$ (5 points)

$10 + 8 \equiv 18 \equiv 5$ mod 13.



7) $7 - 11$ (5 points)

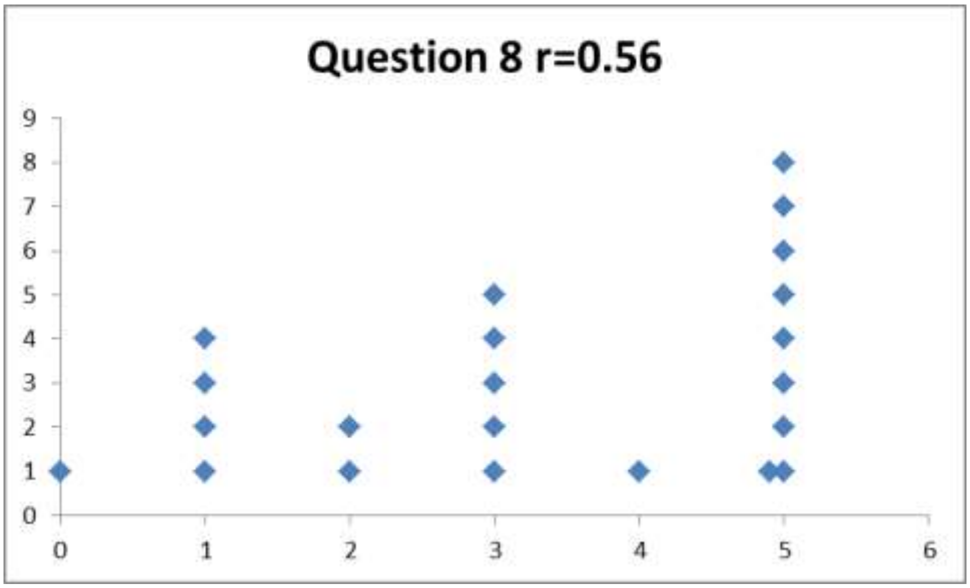
$7 - 11 \equiv -4 \equiv 9 \bmod 13.$



8) Solve $7x \equiv 4 \pmod{13}$. (5 points)

First note that $7^{-1} = 2$ because $2 \cdot 7 \equiv 1$ mod 13.

$7x \equiv 2$ mod 13
$2 \cdot 7x \equiv 2 \cdot 2$ mod 13
$x \equiv 4$ mod 13

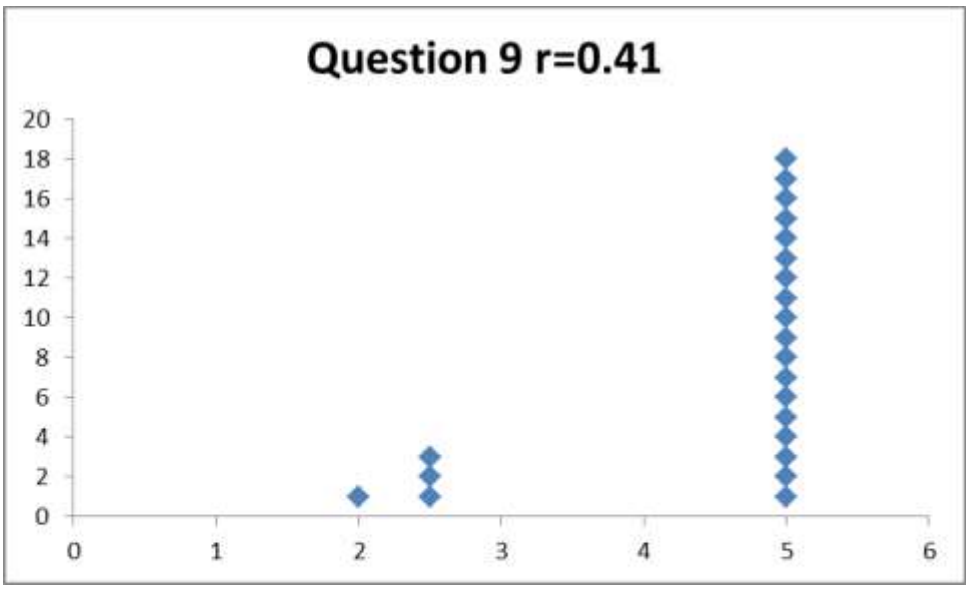


Use the following pseudocode to answer the questions on this page. Assume n is the length of the string s.

```
count = 0
    for i from 0 to n − 1
        if s_i == 'd'
            count = count + 1
            s_i = 'T'
return s, count
```

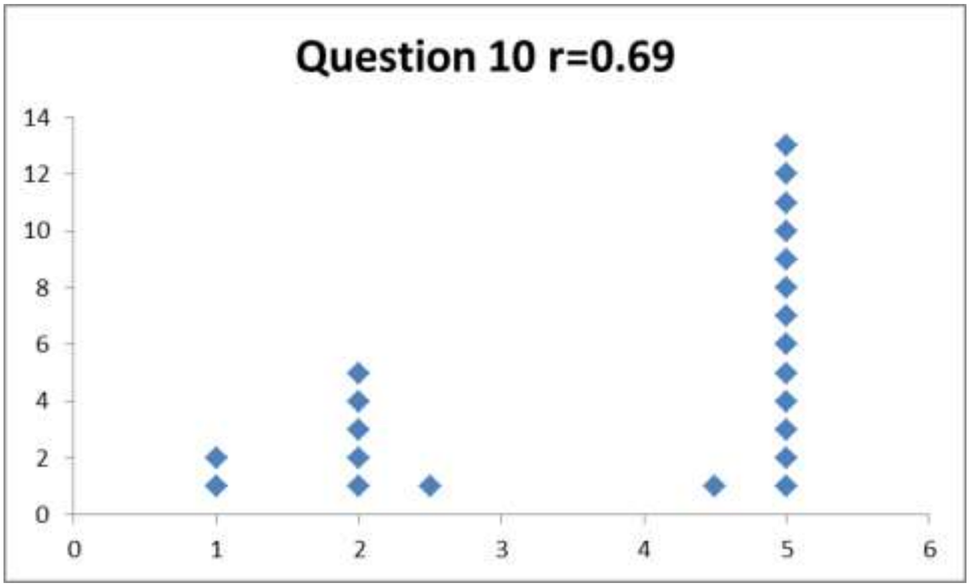
9) For the input s="abcdedcba" what is the return value for count? (5 points)

2



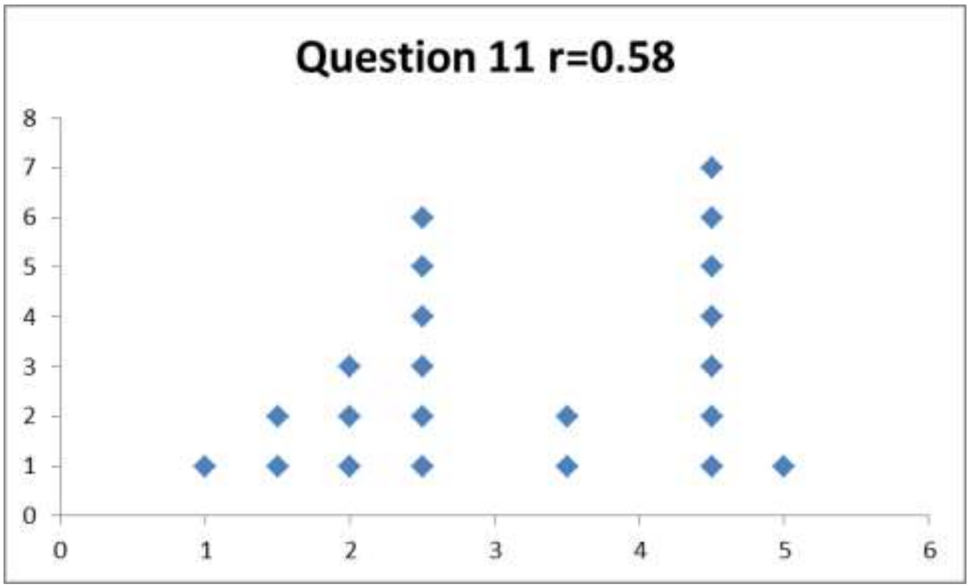
10) For the same input, what is the return value for `s`? (5 points)

“abcTeTcba”



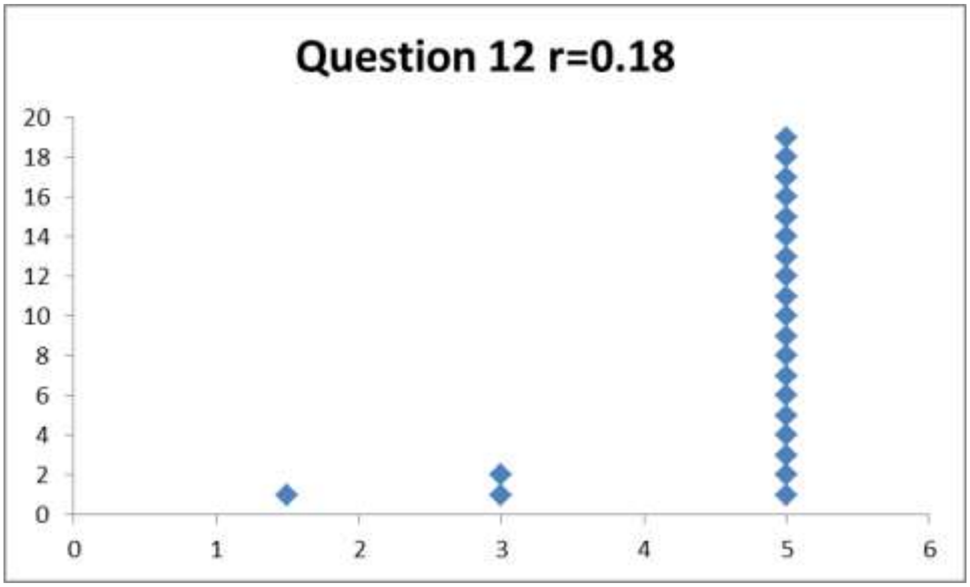
11) For the same input, exactly how many comparisons **of any kind** are performed? (5 points)

19, here's why:
The input has a length of $n = 9$. Hence there are 9 character comparisons: s_i `== 'd'`. There are also a total of 10 integer comparisons between i and $n - 1$: `for i from 0 to n - 1`. The 10th is because 9 are true and the 10th is false.



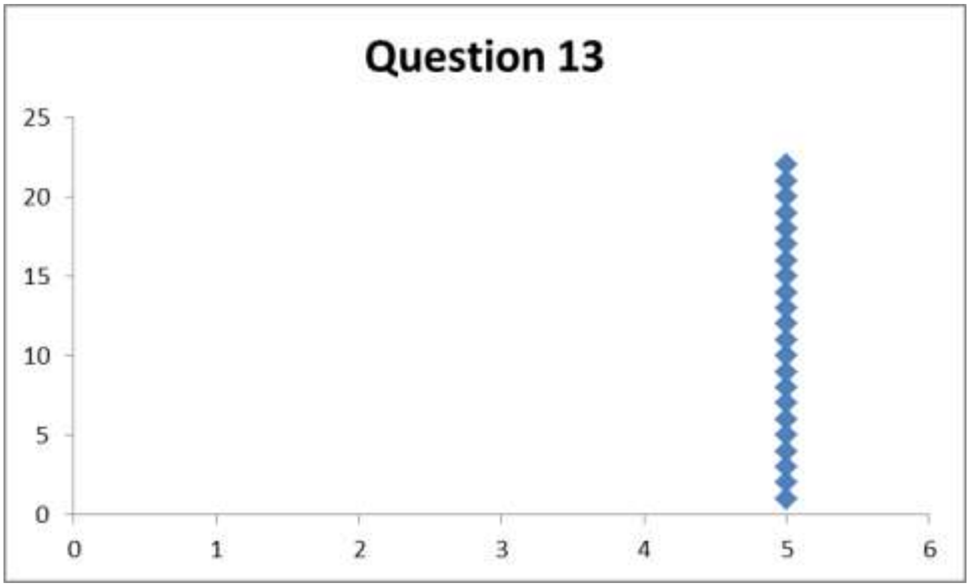
12) For the same input, exactly how many times is `count` assigned a value? (5 points)

3, one for the initialization, and twice when the conditional is true.



13) What is the “big-oh” growth rate of the function $f(n) = 2n^3 + 4n^2 - 5n$? (5 points)

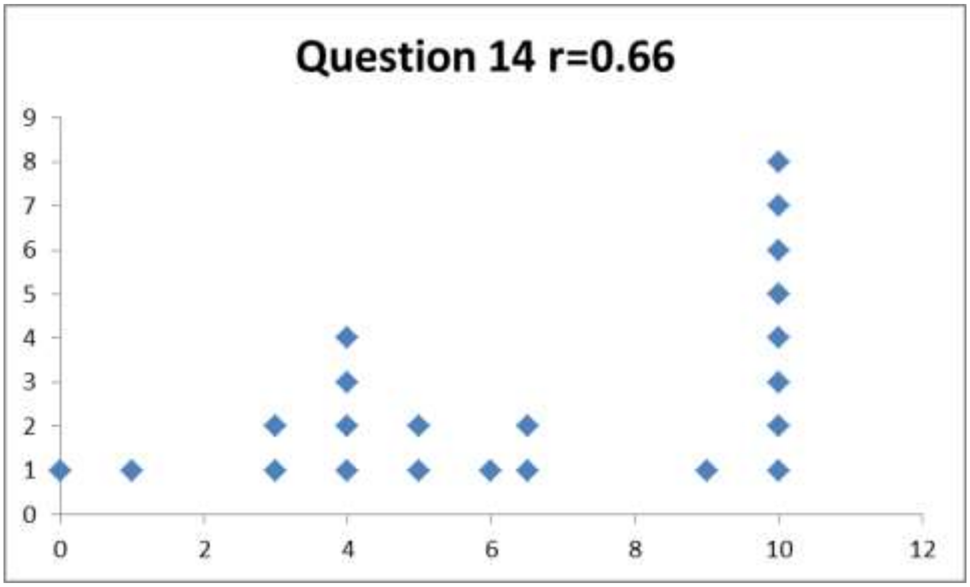
$O(n^3)$



14) Call your answer to the previous question $g(n)$. Justify your answer to the previous by finding the constant multiple and point that it starts to apply: (Fill in the boxes; show and supporting work or derivation below) (10 points)

$$f(n) \leq \boxed{6} \cdot g(n) \text{ whenever } n \geq \boxed{1}$$

$$2n^3 + 4n^2 - 5n \leq 2n^3 + 4n^2 \leq 2n^3 + 4n^3 \leq 6n^3$$

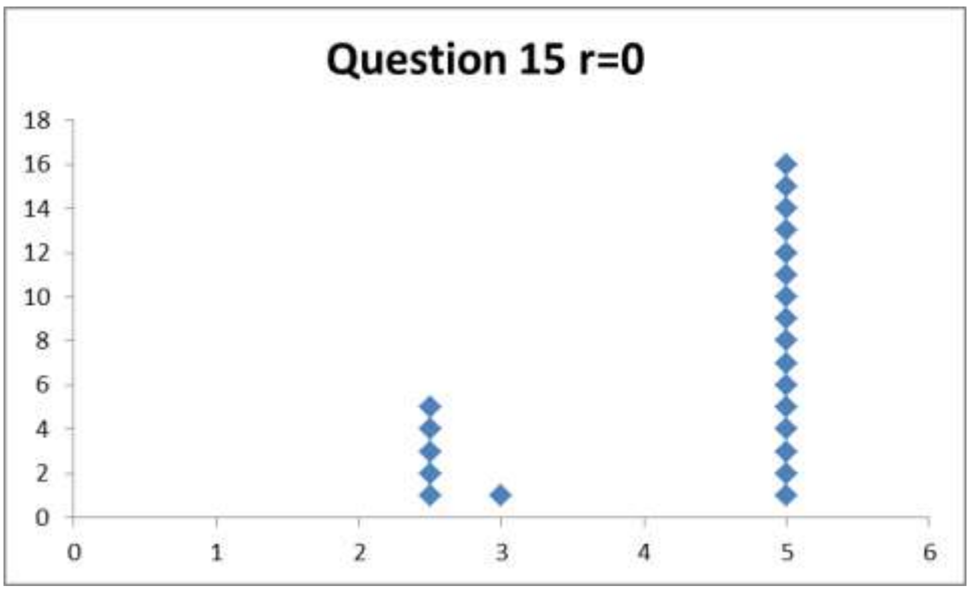


15) Consider the code below. If "`Line 3`" is the line of interest and everything else is trivial, what is the asymptotic growth rate of this algorithm? (5 points)

```
for i from 0 to n-1
   for j from 0 to i
      "Line 3"
```

$$O(n^2)$$

The first loop runs n times. In these n iterations, the second loop runs 1, 2, 3, …, and finally n times. Adding these up we get $\frac{n(n+1)}{2} = O(n^2)$

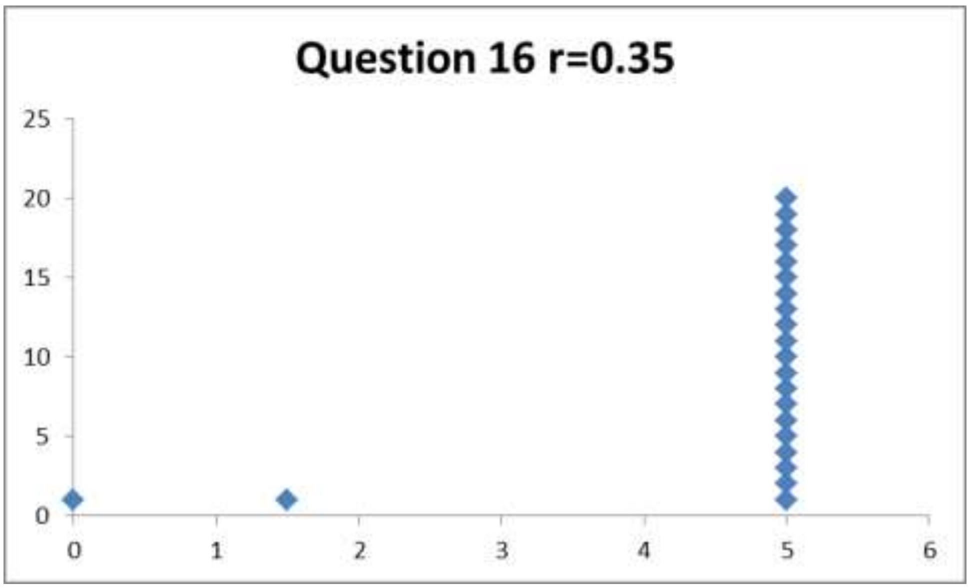


Use the code below to answer questions on this page.

```
x = 1
for i from 0 to n-1
   x = x * 5
```

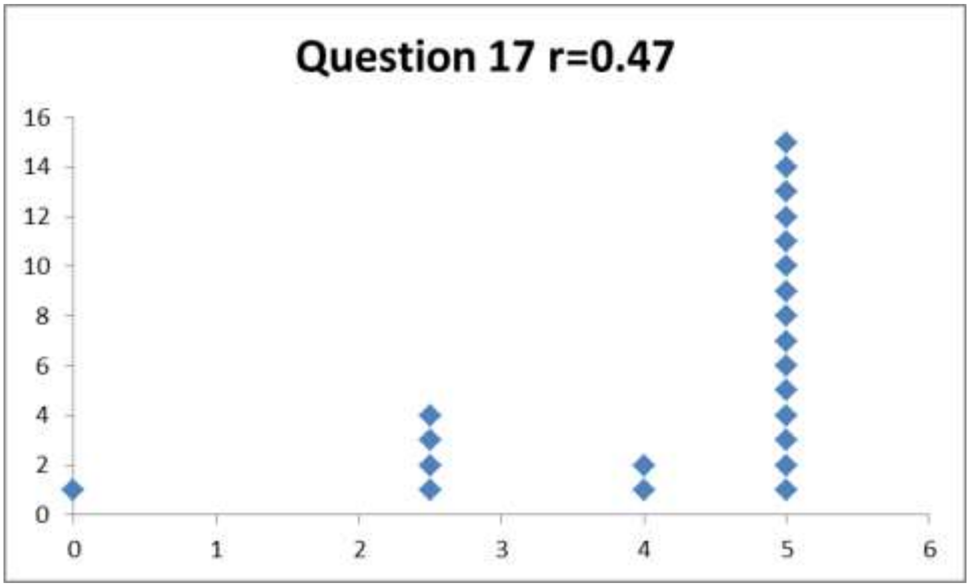
16) Assuming all arithmetic can be done in hardware, what is the asymptotic runtime of this algorithm? (5 points)

$O(n)$



17) Assuming all arithmetic can be done in hardware, what is the asymptotic space requirement of this algorithm? (5 points)

$O(1)$



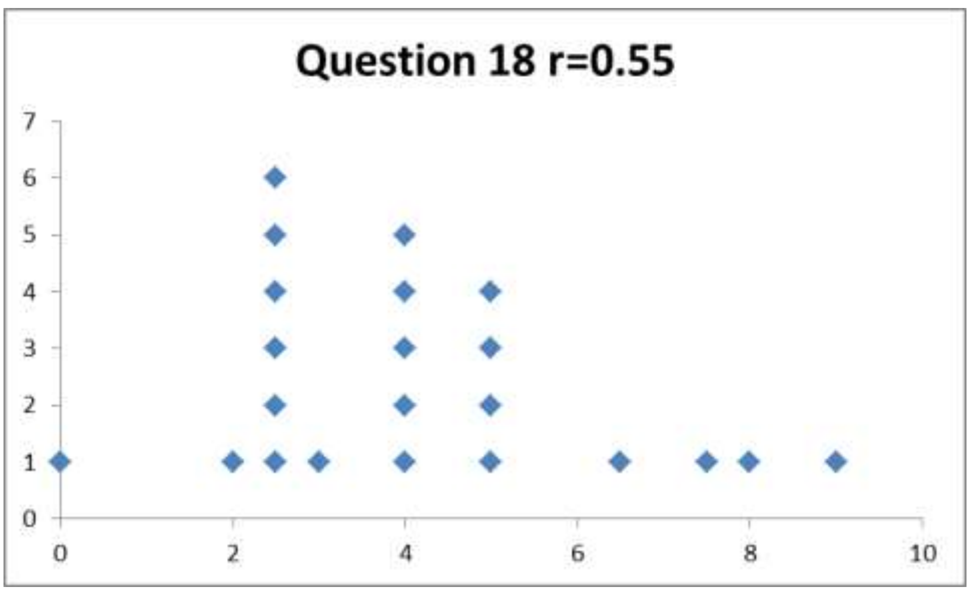
For the next two problems assume that we have a multiplication algorithm that requires $\Theta(m\log(m))$ runtime and $\Theta(\log(m))$ space to multiply two m-bit numbers.
Note that this algorithm doesn't currently exist. It is conjectured that an algorithm with this runtime might exist though, but has not been discovered.

18) If n is large enough that the arithmetic needs to be done in software, what is the asymptotic space requirement of this algorithm?
(10 points – 5 for the answer, 5 for the simplification/derivation)

First note that x gets as large as 5^n. It takes $O(\log(5^n)) = O(n\log(5)) = O(n)$ space. That is x will take $O(n)$ bits just to store. Hence we get:

$$O(n)$$

The additional space required for the multiplication can be reused, so it doesn't change the asymptotic growth rate: $O(n + \log(m)) = O(n)$.



19) If n is large enough that the arithmetic needs to be done in software, what is a bound on the asymptotic runtime of this algorithm?
(10 bonus points)

Again note that x gets as large as 5^n. It takes $O(\log(5^n)) = O(n\log(5)) = O(n)$ space. That is x will take $O(n)$ bits just to store. Hence $m = O(n)$. Then we get that each multiplication requires $O(m\log(m)) = O(n\log(n))$ time. However, there are n multiplications, so we get:

$$O(n^2\log(n))$$