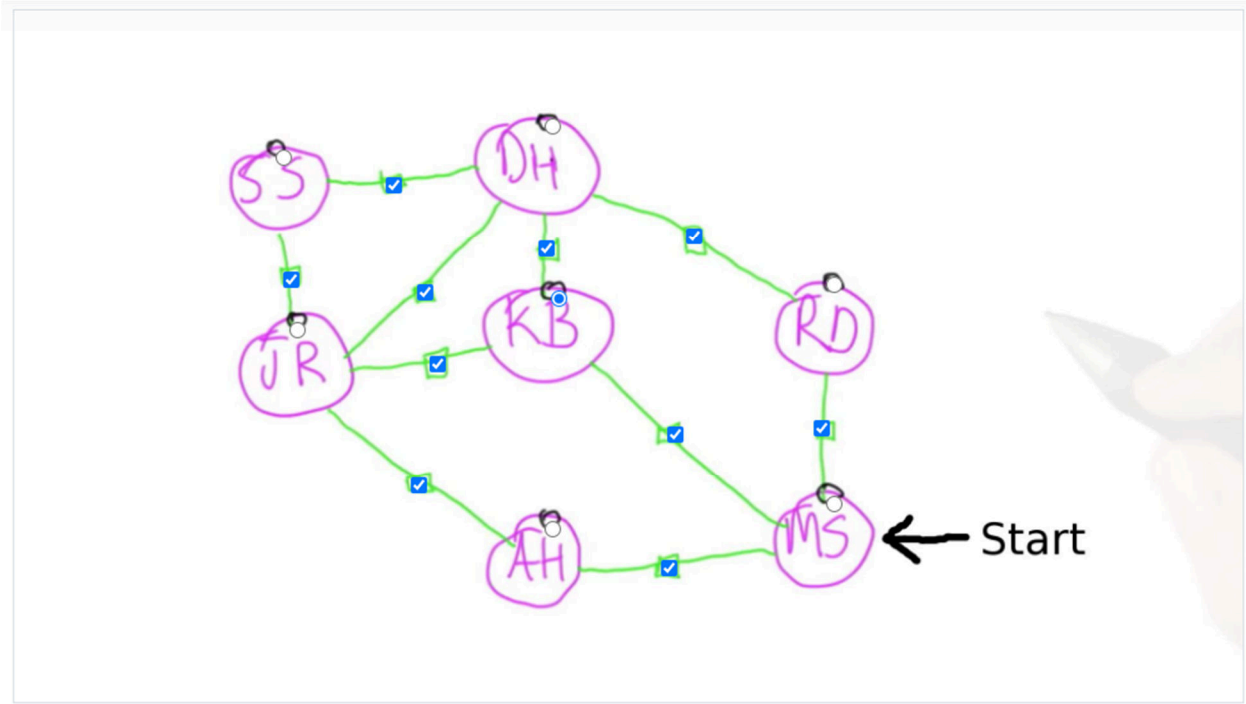
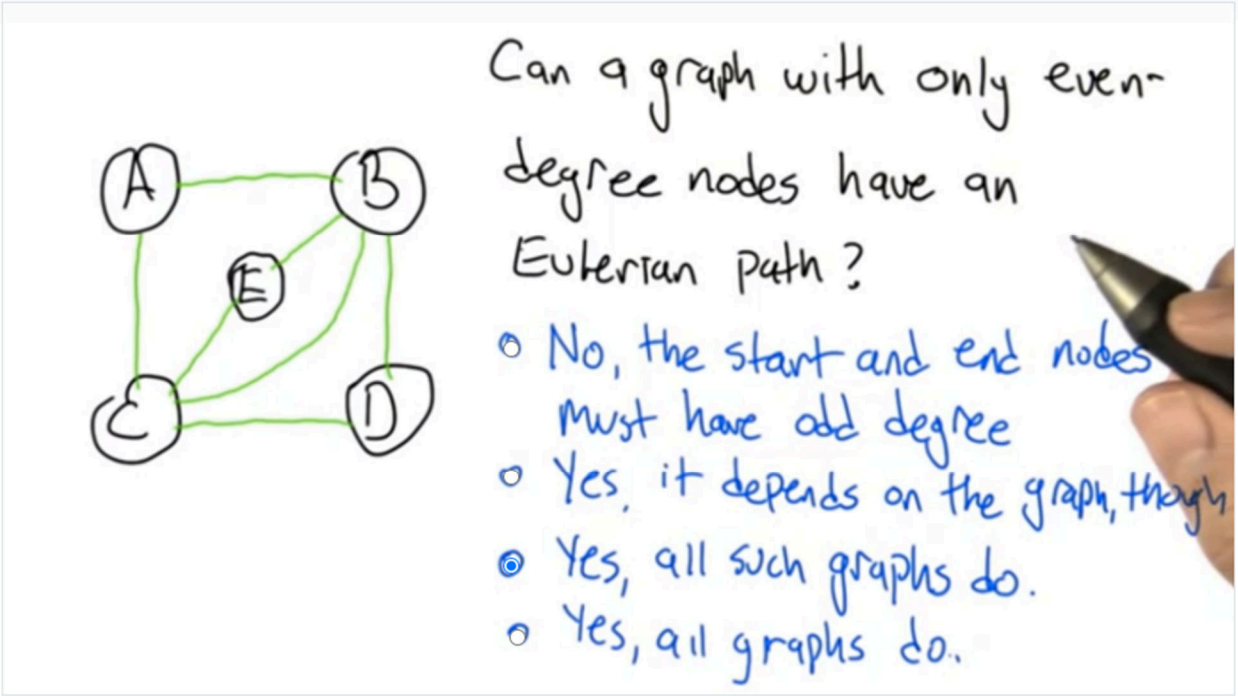
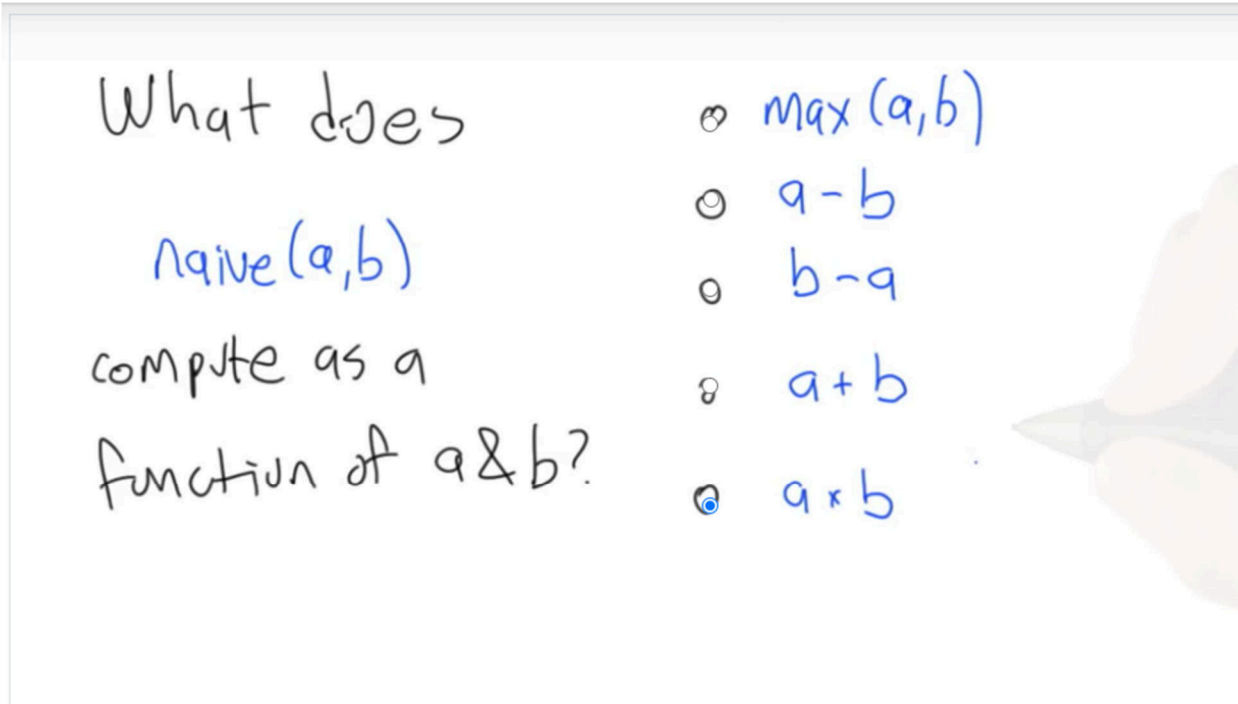
Magic Trick



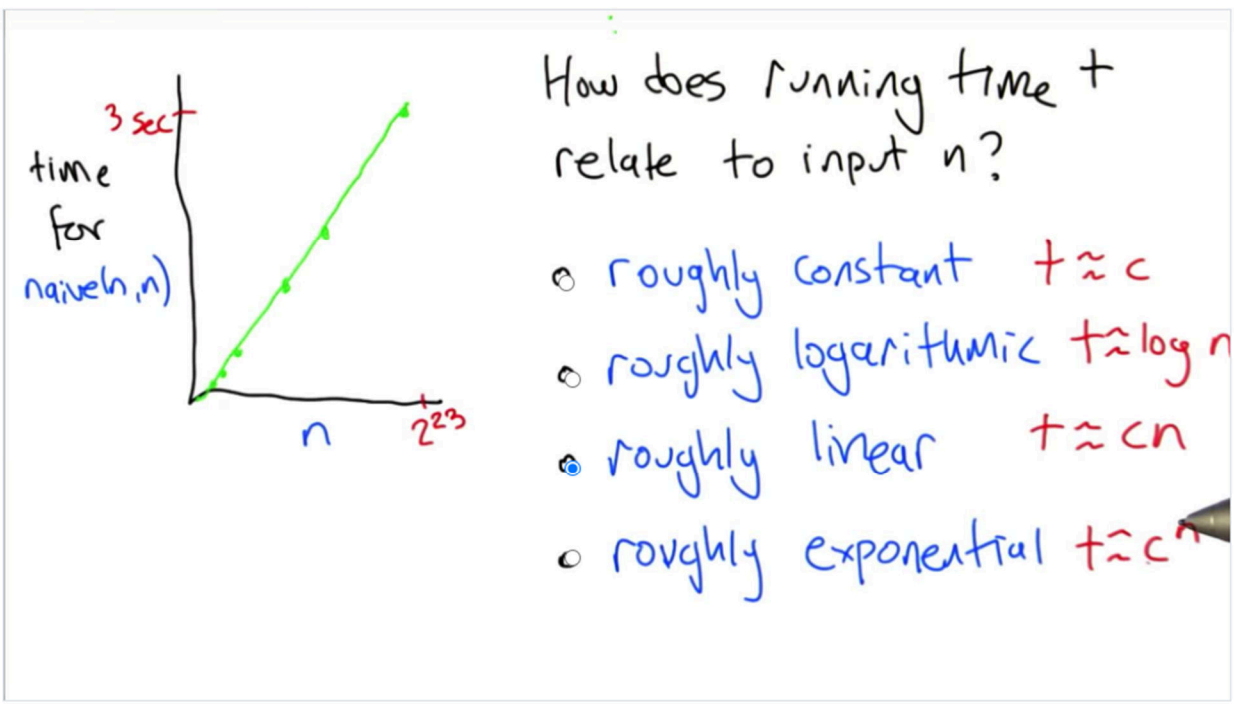
Eulerian Path



Case Study

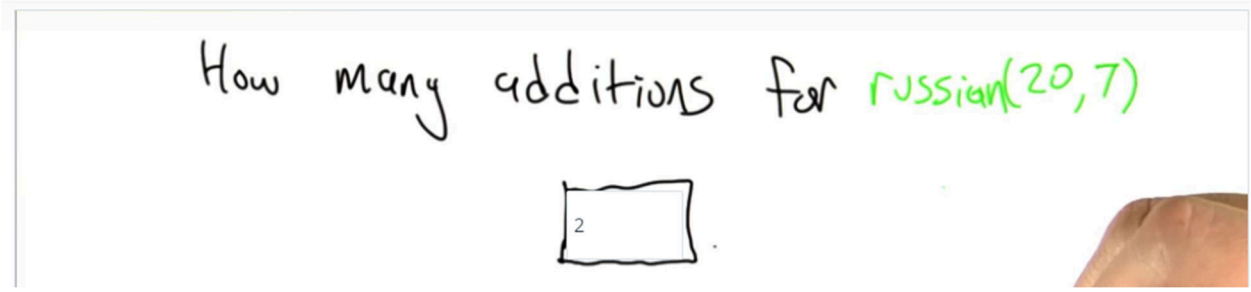


Running Time



Russian Peasant Algorithm

How Many Additions



Measuring Time

```python
# How many units of time will this python code take to run?

s = 0
for i in range(10):
    s = s + i
print s
```

```
12
```

Counting Steps

```python
import math

def time(n):
    """ Return the number of steps
    necessary to calculate
    `print countdown(n)`"""
    steps = 0
    steps = 3 + 2 * math.ceil(n/5.0)
    return steps

def countdown(x):
    y = 0
    while x > 0:
        x = x - 5
        y = y + 1

    return math.ceil(y*2)+3

print countdown(50)
```

Steps for Naïve

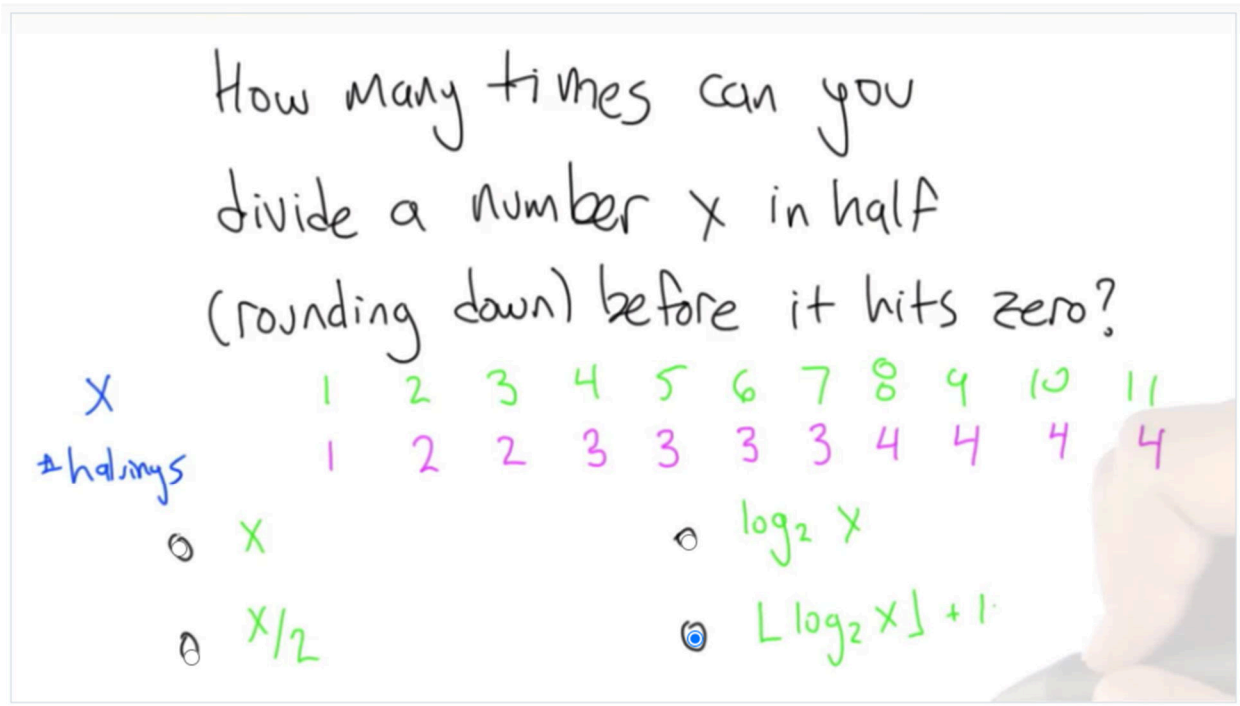
```python
# counting steps in naive as a function of a

def naive(a, b):
    x = a
    y = b
    z = 0
    while x > 0:
        z = z + y
        x = x - 1
    return z

def time(a):
    # The number of steps it takes to execute naive(a, b)
    # as a function of a
    steps = 0
    steps = 2*a +3
    return steps

print (time(5))
```

Halving



Recurrence Relation

$$T(a) = \begin{cases} \text{if } a=0, & 1 \\ \text{elif } a \text{ is even}, & 3 + T(a/2) \\ \text{else}, & 3 + T((a-1)/2) \end{cases}$$

- $T(a) = \lfloor \log_2 a \rfloor + 1$
- $T(a) = 3\lfloor \log_2 a \rfloor + 3$
- $T(a) = 3\lfloor \log_2 a \rfloor + 1$
- $T(a) = 3\lfloor \log_2 a \rfloor + 4$