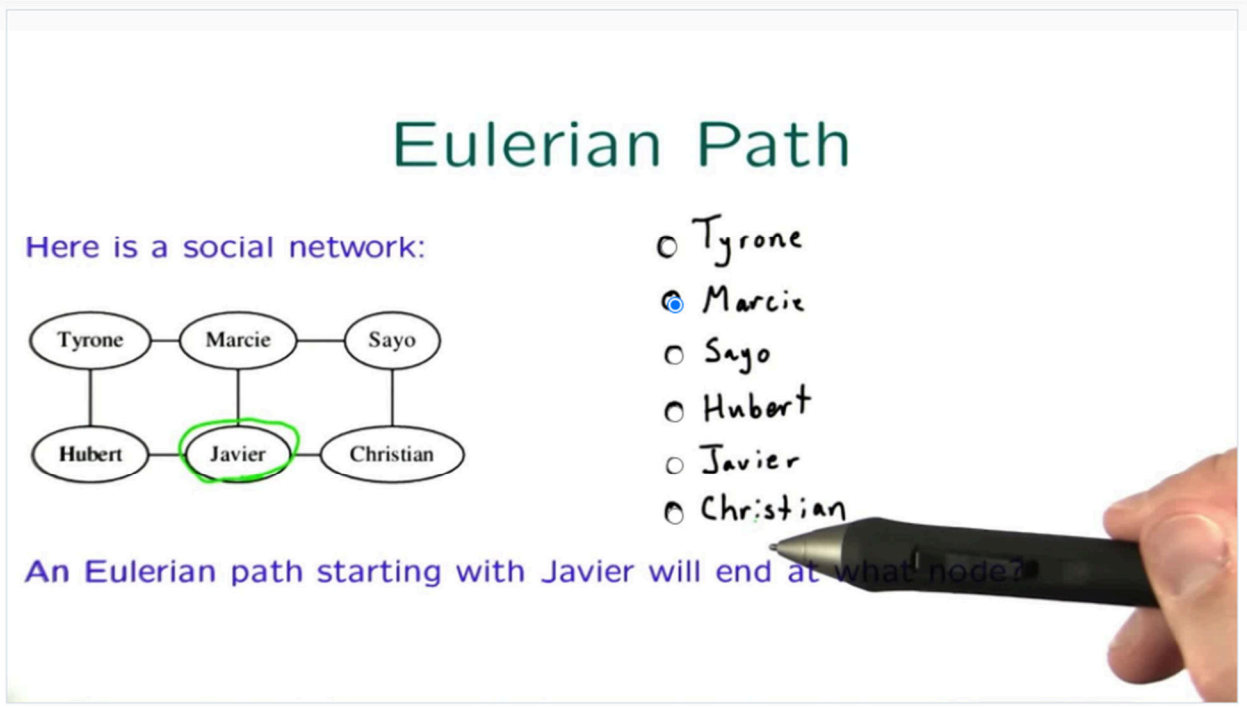
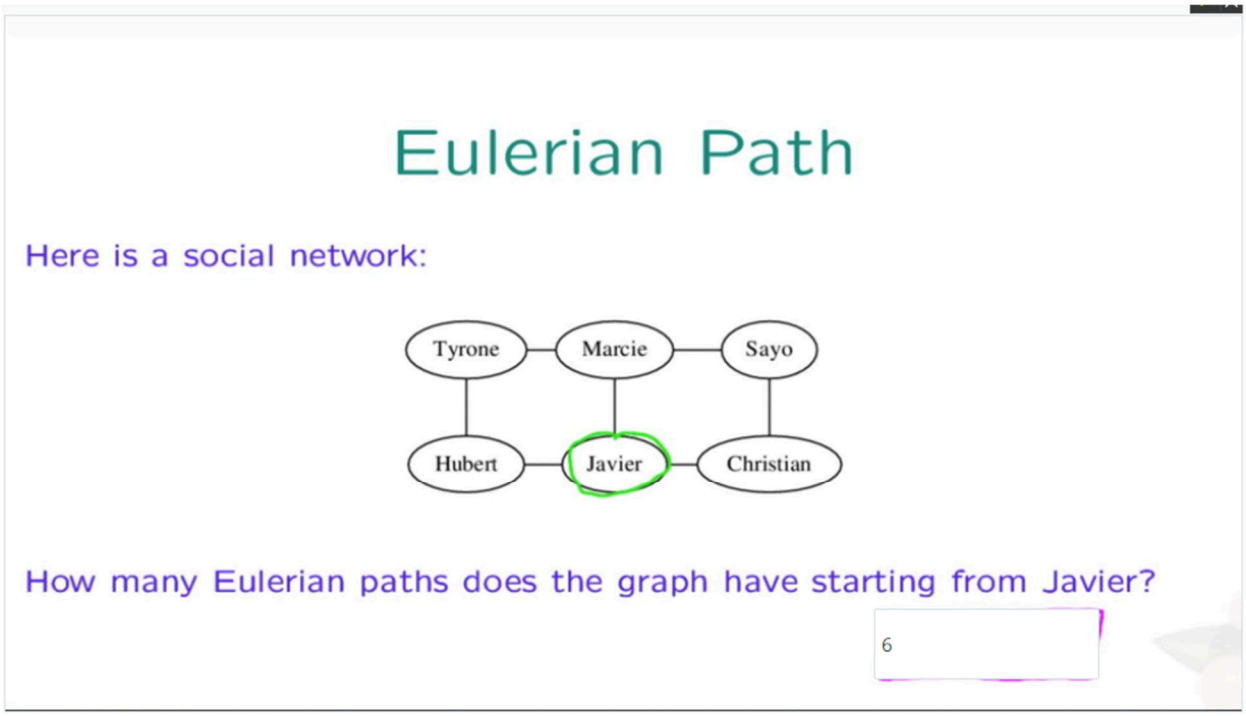
Eulerian Path



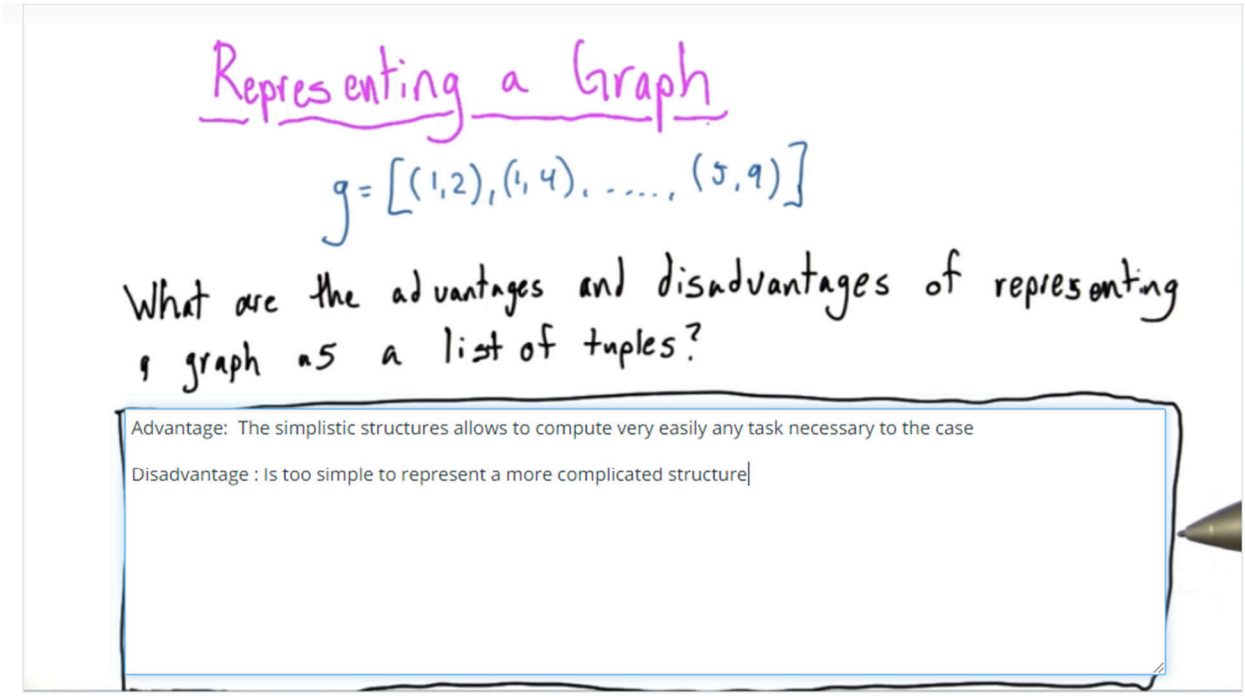
Counting Eulerian Paths



Create Graph with Eulerian Tour

```python
import itertools

def create_tour(nodes):
    # nodes = [1,2,3]
    # return [(1,3), (1,2), (2,3)]
    return list(itertools.permutations(nodes, 2))

#########

def get_degree(tour):
    degree = {}
    for x, y in tour:
        degree[x] = degree.get(x, 0) + 1
        degree[y] = degree.get(y, 0) + 1
    return degree

def check_edge(t, b, nodes):
    """
    t: tuple representing an edge
    b: origin node
    nodes: set of nodes already visited
    if we can get to a new node from `b` following `t`
    then return that node, else return None
    """
    if t[0] == b:
        if t[1] not in nodes:
            return t[1]
    elif t[1] == b:
```

Representing a Graph



Naïve Multiplication Algorithm

Naive Multiplication Algorithm

Recall the algorithm `naive` from lecture.

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

x = 20
y = 12
z = 516

Let's say we're computing `naive(63, 12)`. At some point during the execution we have `x = 20` and `y = 12`. What is `z`?

Recursive Naïve

Recursive Naive Multiplication Algorithm

Here is a recursive version of the naive multiplication algorithm

```
def rec_naive(a, b):
    if a == 0:
        return 0
    return b + rec_naive(a-1, b)
```

How many additions does it take to compute that `rec_naive(17,6) = 102`?

17

Russian Multiplication Algorithm

Russian Multiplication Algorithm

Recall the algorithm russian from the lecture.

```
def russian(a, b):
    x = a; y = b
    z = 0
    while x > 0:
        if x % 2 == 1:
            z = z + y
        y = y << 1
        x = x >> 1
    return z
```

x = 7
z = 84
y = 96

Let's say we're computing `russian(63,12)`. At some point during the execution, we have `x = 7` and `z = 84`. What is `y` at this moment?

Clique

Clique

Here's a loop:

```
def clique(n):
    print "in a clique..."
    for j in range(n):
        for i in range(j):
            print i, "is friends with", j
```

How many units of time does it take to execute `clique(4)`? Count each print statement as one unit and count each time `range` is evaluated as one unit.

12

General Clique

```
1
2  # Write a function, `count`
3  # that returns the units of time
4  # where each print statement is one unit of time
5  # and each evaluation of range also takes one unit of time
6
7  def count(n):
8      # Your code here to count the units of time
9      # it takes to execute clique
10     return 2 + (1+n)*n/2
11
12 def clique(n):
13     print "in a clique..."
14     for j in range(n):
15         for i in range(j):
16             print i, "is friends with", j
17
18 if __name__ == '__main__':
19     print count(4)
```

Challenge Find Eulerian Tour

```
def find_eulerian_tour(graph):

    def _next_node(edge, current):
        return edge[0] if current == edge[1] else edge[1]

    def _remove_edge(raw_list, discard):
        return [item for item in raw_list if item != discard]

    search = [[[], graph[0][0], graph]]
    while search:
        path, node, unexplore = search.pop()
        path += [node]

        if not unexplore:
            return path

        for edge in unexplore:
            if node in edge:
                search += [[path, _next_node(edge, node), _remove_edge(unexplore, edge)]]

if __name__ == '__main__':
    graph = [(1, 2), (2, 3), (3, 1), (3, 4), (4, 3)]
    print find_eulerian_tour(graph)
```