CS 189 today

A few more problems... ... and one more algorithm!

Open-ended project option...

Jotto game finale?!

Rest of the term

*This is our last meeting!*

You may submit problems up until graduation...

*Unless you're a senior – then you have until 5/9*
This term's first class to guess another's word earns 1 problem...
This term's last class to have its word guessed earns 1 problem...
**Jotto so far!**

Each team may need to **create** a valid word (easier than remembering it!)

(3) [*jotto*] Not really a to-do item, but just in case it reminds folks of their hidden word new, compatible ones -- Ben H. has requested the jotto scores thus far. Here they are -

<table>
<thead>
<tr>
<th></th>
<th>diner</th>
<th>bloat</th>
<th>gumbo</th>
</tr>
</thead>
<tbody>
<tr>
<td>FYears (Dan/Sam):</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SophSelectTeam:</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SophFunTeam:</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>JRs:</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SRs:</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExtraMudd:</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>&quot;Other&quot; (me):</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

*This term's first class to guess another's word earns 1 problem...*

*This term's last class to have its word guessed earns 1 problem...*
Last couple of weeks... (!)

Facebook Values WhatsApp Users at $42 Each

According to this SEC filing, Facebook (FB +0.4%) is buying messaging app WhatsApp for an incredible $19 billion dollars: $12 billion in stock, $4 billion in cash and another $3 billion in restricted stock units.

At last count of the company’s 55 staff, that’s a cool $345 million per employee.

Time for some CS tourism!
presenting...
Stanford...

himself!?
CS tourism!
Earlier conference ~ poster session...
Earlier conference ~ poster session...
An Interactive Classroom Management Tool: ClassPi

Motivation

Student-Teacher Interactions
- The need for students to communicate their thoughts and ideas to the instructor directly
- To create a venue for real-time student-teacher communication, without interrupting the class

College students are bored in class about 28% of the time.

Group Motivation
- Our desire to engage in a project where several different skill levels are necessary to succeed

The Current State of Technology in the Classroom (Infographic)
Attending posters...

Student-Teacher Interactions

- The need for students to communicate their thoughts and ideas to the instructor directly
- To create a venue for real-time student-teacher communication, without interrupting the class

College students are bored in class about 28% of the time.

I think it's the 28% of the time that the instructor goes on irrelevant tangents about recent travels!!
Optional open-ended project: *April*

- worth up to +8 problems ~ also, an opportunity...
- ... to try out / get familiar with / learn about a *technology, domain, library, or project*

**Plan:**

1. (0) decide what you'd like to learn...
2. (1) find a reasonable resource for it...
3. (2) create a project and a write-up...
4. (3) time expectation: 3 hours per week
Max Flow!

What's the maximum flow possible, from src to sink?

Ford-Fulkerson algorithm
Max Flow

(Step #1) Use depth- or breadth-first search to find any path from s to t.

What's left?
Max Flow

(Step #2) Compute RESIDUAL graph

What's left... Residual capacities.

and the red edges?

Backwards capacities!
Max Flow

(Step #3) Repeat until no path exists...

Residual capacities.
Backwards capacities.
Max Flow

(Step #1) Use depth- or breadth-first search to find any path from s to t.

(Step #2) Compute RESIDUAL graph

(Step #3) Repeat until no path exists...

max flow: 23
Python...

```python
if __name__ == "__main__":

    # make a capacity graph
    # node   A   B   C   D   E   F
    C = [[00, 16, 13, 00, 00, 00],  # A
         [00, 00, 10, 12, 00, 00],  # B
         [00, 04, 00, 00, 14, 00],  # C
         [00, 00, 9, 00, 00, 20],   # D
         [00, 00, 00, 7, 00, 4 ],   # E
         [00, 00, 00, 00, 00, 00]]  # F

    print "C is", C
    source = 0  # A
    sink = 5    # F

    max_flow_value = max_flow( C, source, sink )
    print "max_flow_value is", max_flow_value
```

Linked at the ACM website by the slides...
def max_flow(C, source, sink):
    n = len(C)  # C is the capacity matrix
    F = [[0] * n for i in range(n)]  # F is the flow matrix
    # residual capacity from u to v is C[u][v] - F[u][v]

    while True:
        path = BFS(C, F, source, sink)
        if not path:
            break  # no path - we're done!

        edges = [C[u][v]-F[u][v] for u,v in path]
        path_flow = min(edges)

        print "Augmenting by", path_flow
        for u,v in path:  # traverse path to update flow
            F[u][v] += path_flow  # forward edge up
            F[v][u] -= path_flow  # backward edge down

    return sum([F[source][i] for i in range(n)])  # out from source
def BFS(C, F, source, sink):
    queue = [source]  # the BFS queue
    paths = {source: []}  # stores 1 path per graph node
    while queue:
        u = queue.pop(0)  # next node to explore (expand)
        for v in range(len(C)):
            # path from u to v? and not yet at v?
            if C[u][v] - F[u][v] > 0 and v not in paths:
                paths[v] = paths[u] + [(u,v)]
                if v == sink:
                    return paths[v]
        queue.append(v)  # go from v in the future
    return None
Is **maxflow** good for anything *else*?

that is, beyond solving maximum-flow problems...
we have four brides and six grooms

and some acceptable possibilities ...

Matching!

a bipartite graph
we have four brides and six grooms

and some acceptable possibilities ...

Matching!

a maximal matching == no more matchings without rearrangement
Matching! and *some* acceptable possibilities ...

we have four brides and six grooms

![Diagram showing a maximum matching](https://via.placeholder.com/150)

*a maximum matching* == no rearrangements will yield more matchings
Maximum matching *is* max flow...

connect a source to the left side...

![Diagram](source)

- **S**: Source
- **all 1s**
Maximum matching is max flow...

cconnect a source to the left side...

make all capacities = 1

all 1s

source
Maximum matching *is* max flow...

- Connect a source to the left side...
- Make all capacities = 1
- Put a sink on the right

All 1s

Source

Sink

What do the source and sink constraints ensure?
Max flow thought experiment...

Suppose this is the flow so far (3 units):

Draw what happens in the next step of the max-flow algorithm!

how to get from *maximal* matching to *maximum* matching...
Max flow thought experiment...

... the path it finds ...

What's going on here?
Max flow thought experiment...

Done!

Maximum matching == 4
This week's problems...

dinner
dining
hardware
muddy
feeding

all can be done with maxflow...
The challenge:

is often *setting up* the graph
There are four tools available — 

- hammer
- phone
- coffee
- laptop

at these costs:

- hammer: 44
- phone: 42
- coffee: 10
- laptop: 1000

There are four tasks available — 

- hammer
- phone
- coffee
- PC

with these rewards:

- hammer: 50
- phone: 20
- coffee: 8
- PC: 3

Each task requires some tools:

- PHP coding
- sleep

What is flowing?
How do we use the results?
There are four tools available ~ at these costs:
- hammer
- phone
- coffee
- laptop

There are four tasks available ~ with these rewards:
- E4
- PHP coding
- sleep
- coding

What is flowing?
How do we use the results?
There are four tools available ~ at these costs:
- hammer
- phone
- coffee
- laptop

Each task requires some tools.

There are four tasks available ~ with these rewards:
- E4
- PHP coding
- sleep
- Netflix

What is flowing? How do we use the results?
Input

number of teams
4 5
4 5 3 5
3 5 2 6 4
4 5
4 5 3 5
3 5 2 6 3
0 0

number of tables

# of people in each team

capacity of each table

Output

1 0
can an assignment be made without putting teammates together?

tenants with sizes

3 5
2 6 4

10

capacity of each table

again...

capacity of each table

again...

can an assignment be made without putting teammates together?

tables with capacities

3 5
2 6 4

end...
How do these edge weights reflect the problem constraints?

How does maxflow help?
What?!

Oddities from computer code...
original *Wat* talk...
php -a

php > $x = "209";

php > $x++;

php > print( $x ); print( "\n" );

210
php -a

php > $x = "209";
php > $x++;
php > print( $x ); print( "\n" );
210

php > $x = "may";
php > $x++;
php > print( $x ); print ( "\n" );
maz
php > $x++;
php > print( $x ); print ( "\n" );
mba

Now, let's try it with...

$x = "2d9";
1 == True

0 == False

(2==2) == 2
(1==1) == 1
0== (0==0)
0== (1==0)
1==1==1
0==0==0

Are these True or False?
Good luck through April!

Jotto guess!

April is the cruellest month...

T.S. Eliot
Jotto so far!

(3) [*jotto*] Not really a to-do item, but just in case it reminds folks of their hidden word new, compatible ones -- Ben H. has requested the jotto scores thus far. Here they are -

<table>
<thead>
<tr>
<th></th>
<th>diner</th>
<th>bloat</th>
<th>gumbo</th>
</tr>
</thead>
<tbody>
<tr>
<td>FYears (Dan/Sam):</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SophSelectTeam:</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SophFunTeam:</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>JRs:</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>SRs:</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ExtraMudd:</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>&quot;Other&quot; (me):</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

each team may need to **create** a valid word (easier than remembering it!)

*Guesses!*?

This term's first class to guess another's word earns 1 problem...
This term's last class to have its word guessed earns 1 problem...