CS 189 today

A few more problems… … and one more algorithm!

Open-ended project option…

First ones have been scored…

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

Jotto... so far!

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

Sophs	JRs	SRs	POM-CMC-SCR-PTZ	*other*
2	**1**	**1**	**0**	**1**
diner 1	**diner 1**	**diner 1**	**diner 1**	**diner 1**
0	**1**			
bloat 3	**bloat 1**	**bloat 0**	**bloat 1**	**bloat 1**
0	**0**			
gumbo 0	**gumbo 0**	**gumbo 0**	**gumbo 2**	**gumbo 0**

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 -

	diner	bloat	gumbo
FYears (Dan/Sam):	2	0	0
SophSelectTeam:	1	1	0
SophFunTeam:	1	3	0
JRs:	1	1	0
SRs:	0	0	0
ExtraMudd:	1	1	2
"Other" (me):	1	1	0

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… (!)

THE WALL STREET JOURNAL. ≡ BUSINESS

C. Corporate Intelligence

What matters right now in business. From W world.

TECHNOLOGY | MEDIA | AUTOS | REGULA

5:17 pm
Feb 19, 2014 MERGERS

Facebook Values WhatsApp Users at $42 Each

ARTICLE | COMMENTS (2)

BUSINESS FACEBOOK MOBILE WHATSAPP

Email Print

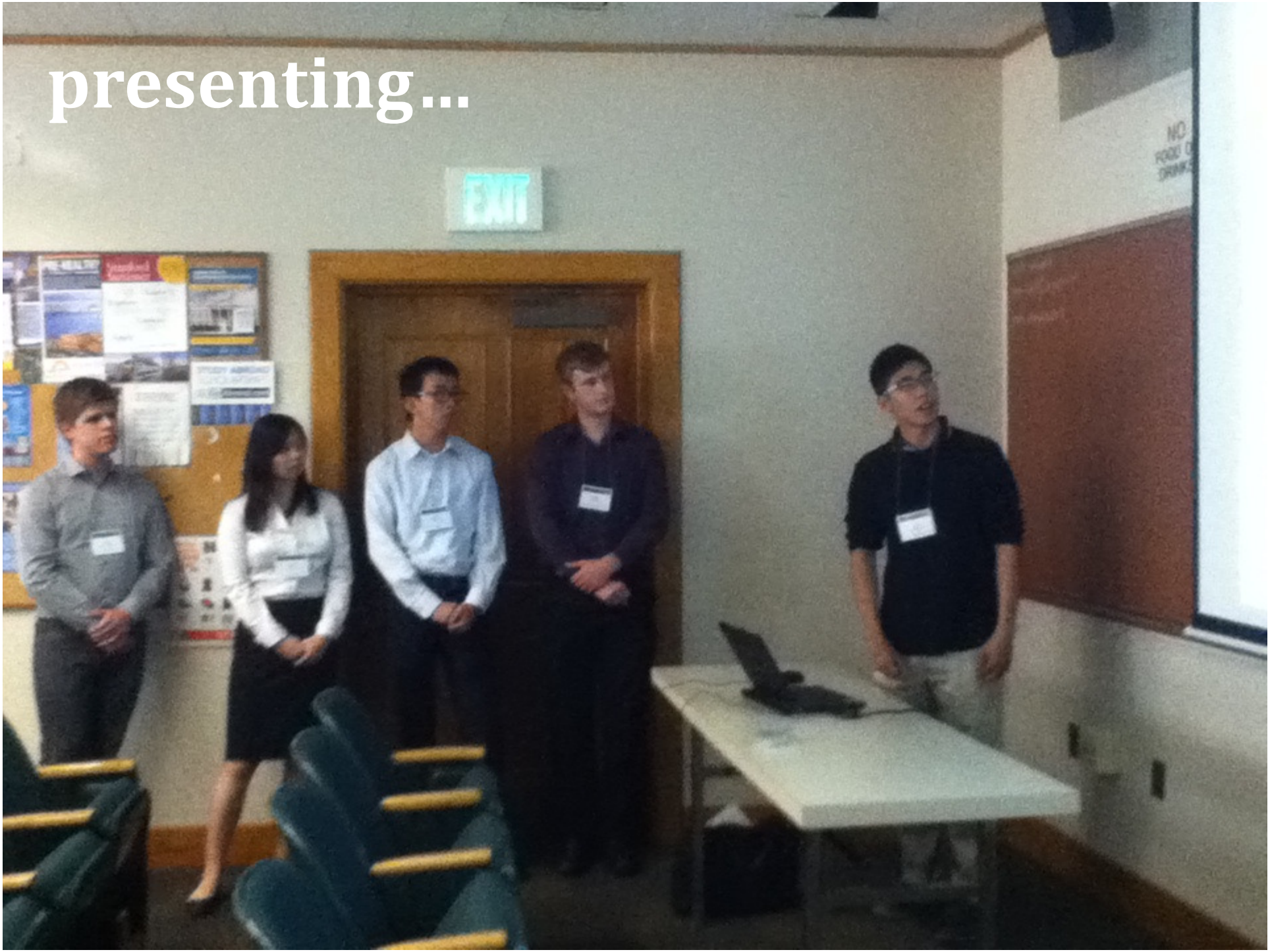
By TOM GARA CONNECT

According to this SEC filing, Facebook FB +0.45% 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!?

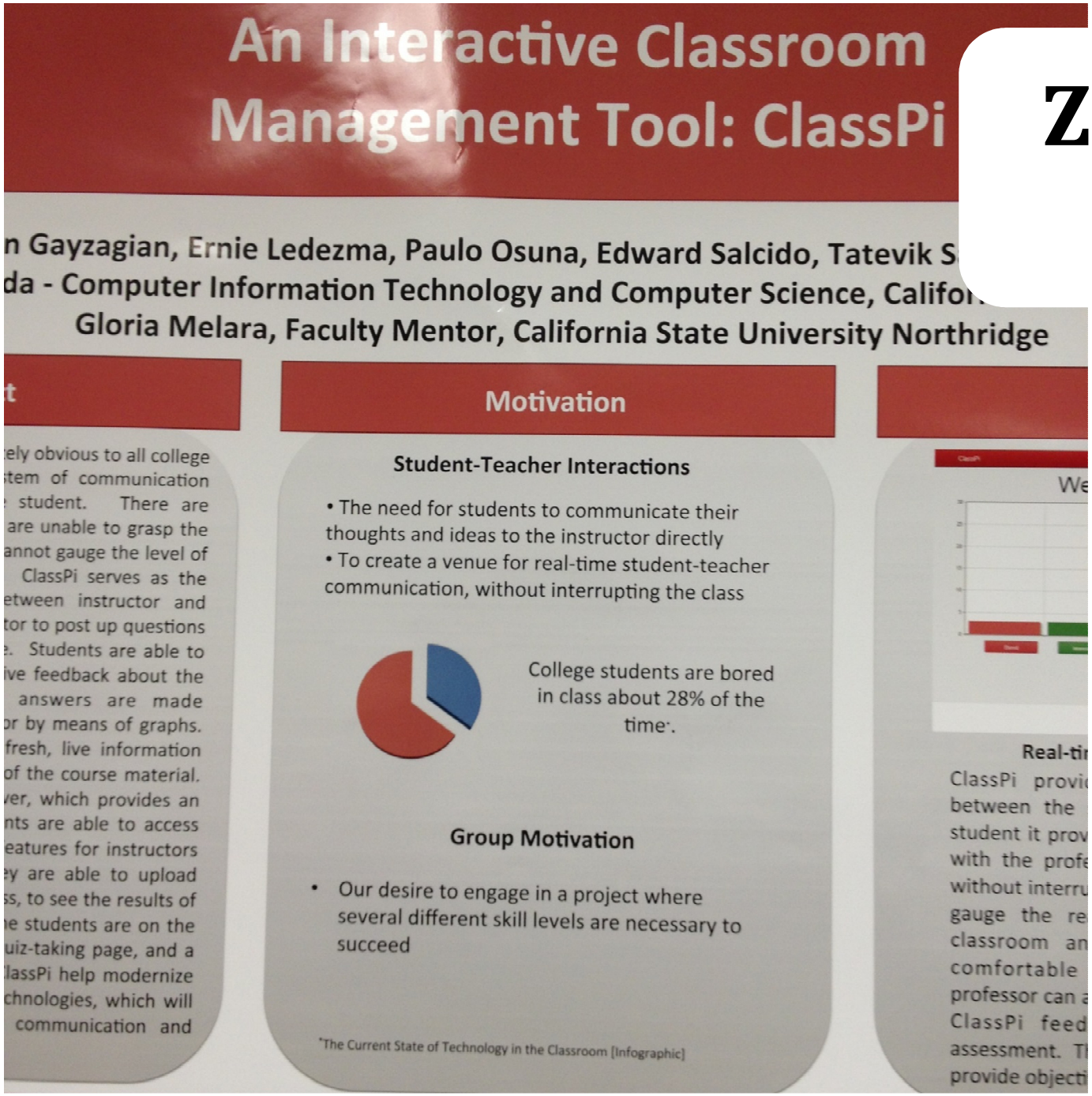
CS tourism!

Google

Earlier conference ~ poster session…

Earlier conference ~ poster session…

Zooming in…



An Interactive Classroom Management Tool: ClassPi

n Gayzagian, Ernie Ledezma, Paulo Osuna, Edward Salcido, Tatevik S
da - Computer Information Technology and Computer Science, Califor
Gloria Melara, Faculty Mentor, California State University Northridge

t

ely obvious to all college
stem of communication
student. There are
are unable to grasp the
annot gauge the level of
ClassPi serves as the
etween instructor and
tor to post up questions
. Students are able to
ive feedback about the
answers are made
or by means of graphs.
fresh, live information
of the course material.
ver, which provides an
nts are able to access
eatures for instructors
y are able to upload
ss, to see the results of
e students are on the
uiz-taking page, and a
lassPi help modernize
chnologies, which will
communication and

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*.

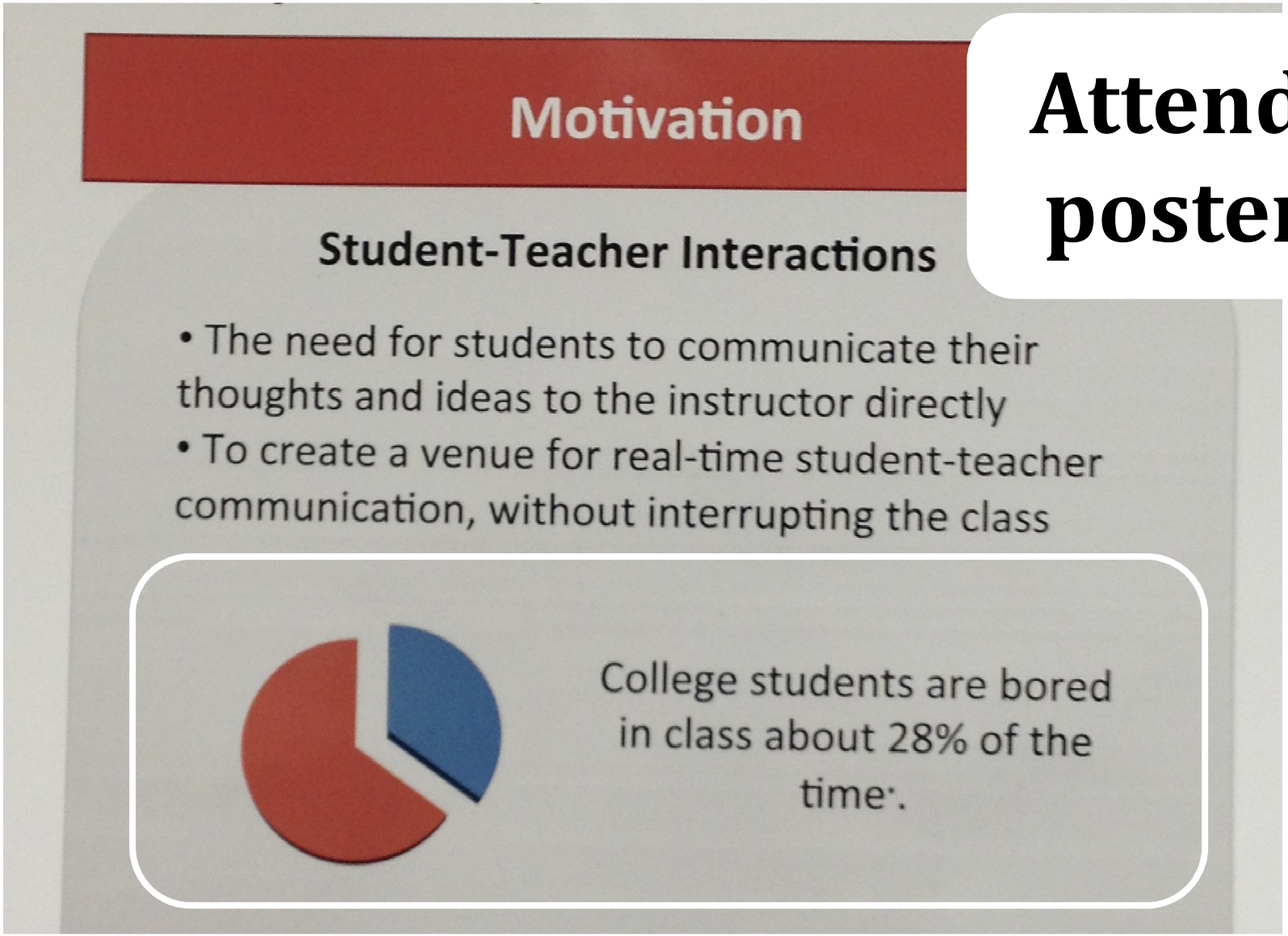
Group Motivation

- 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]

Real-ti
ClassPi provi
between the
student it prov
with the profe
without interru
gauge the re
classroom an
comfortable
professor can a
ClassPi feed
assessment. Th
provide objecti

Attending posters…



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:

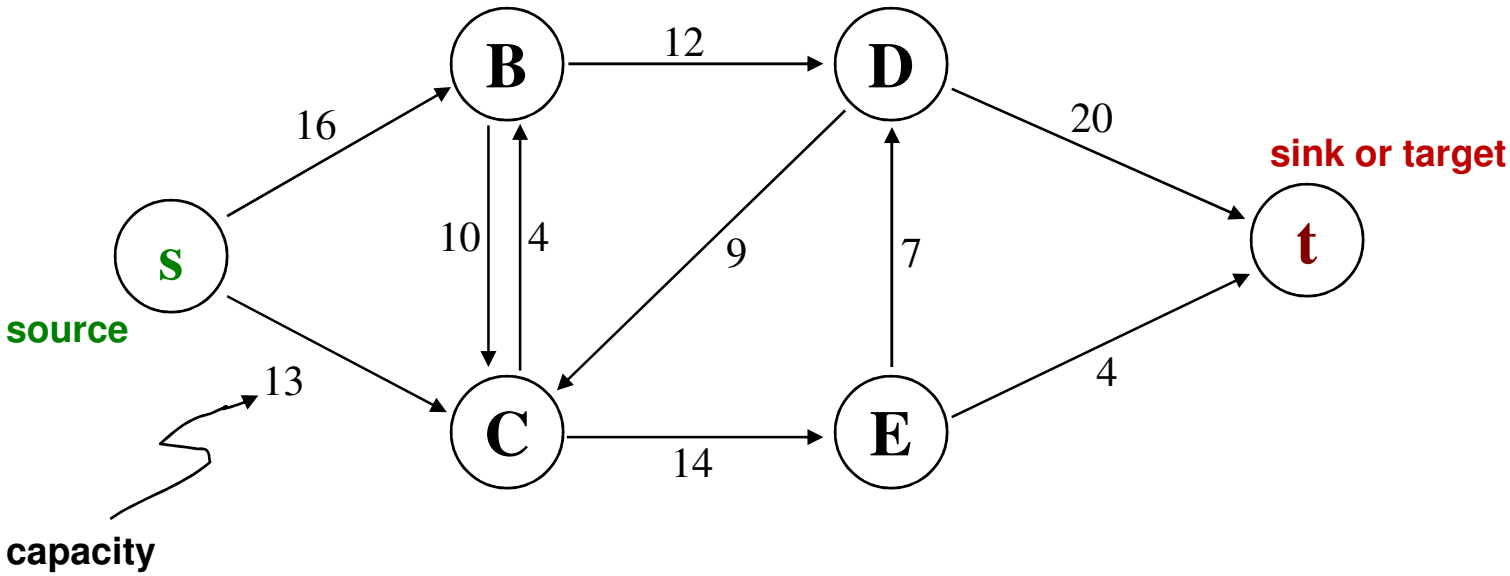
(0) decide what you'd like to learn…

(1) find a reasonable resource for it…

(2) create a project and a write-up…

(3) time expectation: 3 hours per week

Max Flow !



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

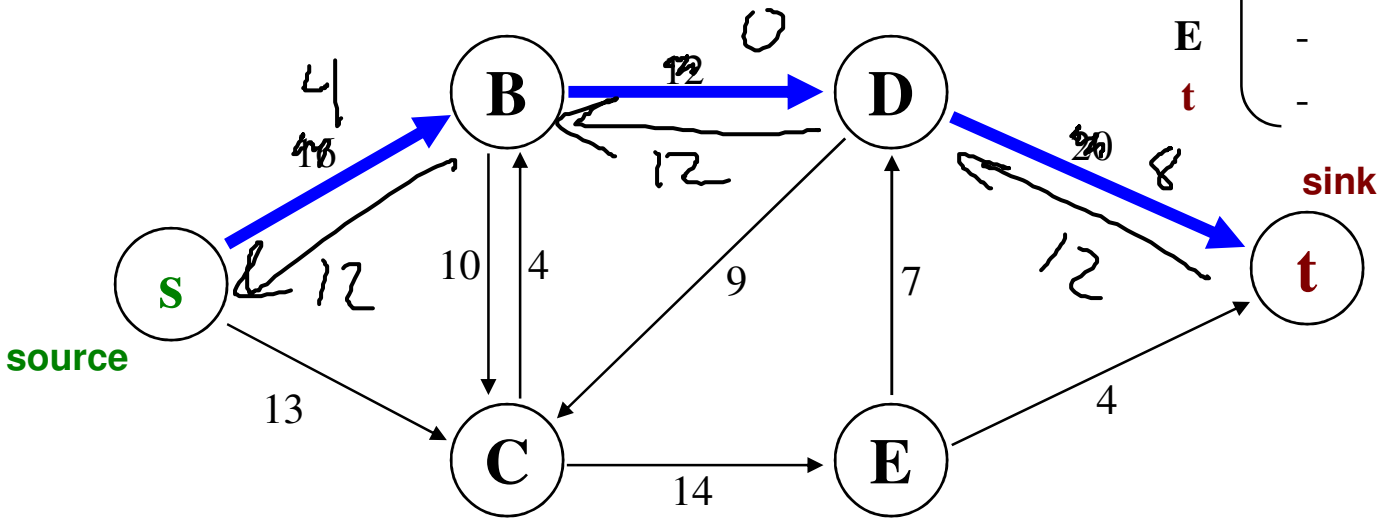
Ford-Fulkerson algorithm

Max Flow

Capacity Graph

FROM \ TO	s	B	C	D	E	t
s	-	16	13	-	-	-
B	-	-	10	12	-	-
C	-	4	-	-	14	-
D	-	-	9	-	-	20
E	-	-	-	7	-	4
t	-	-	-	-	-	-

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

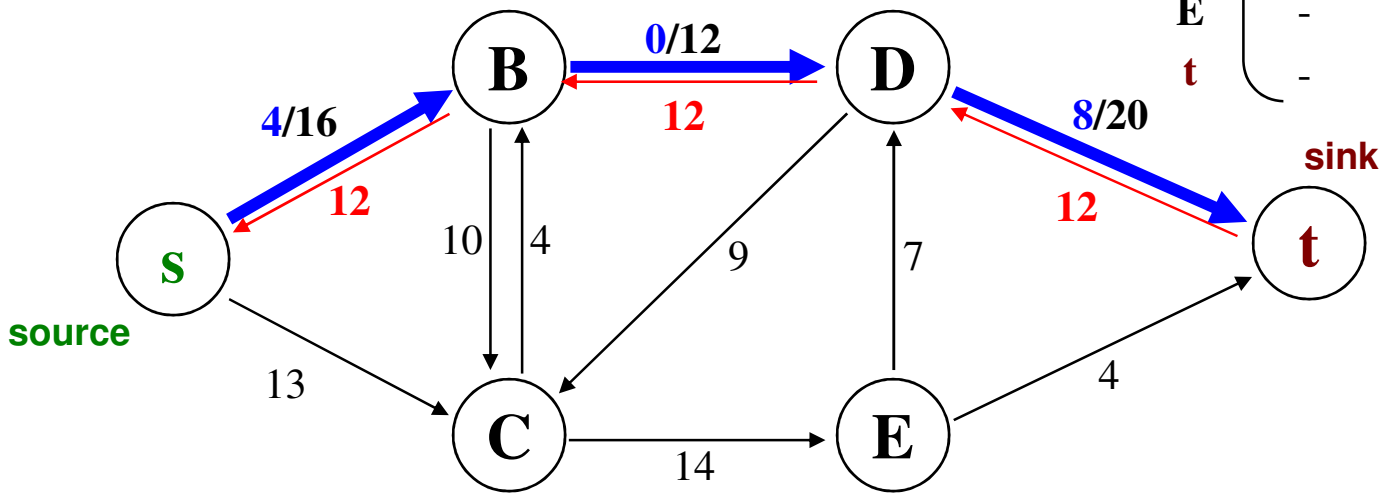


What's left ?

Max Flow

(Step #2) Compute RESIDUAL graph

	s	B	C	D	E	t
s	-	16	13	-	-	-
B	-	-	10	12	-	-
C	-	4	-	-	14	-
D	-	-	9	-	-	20
E	-	-	-	7	-	4
t	-	-	-	-	-	-



What's left… *Residual capacities.*

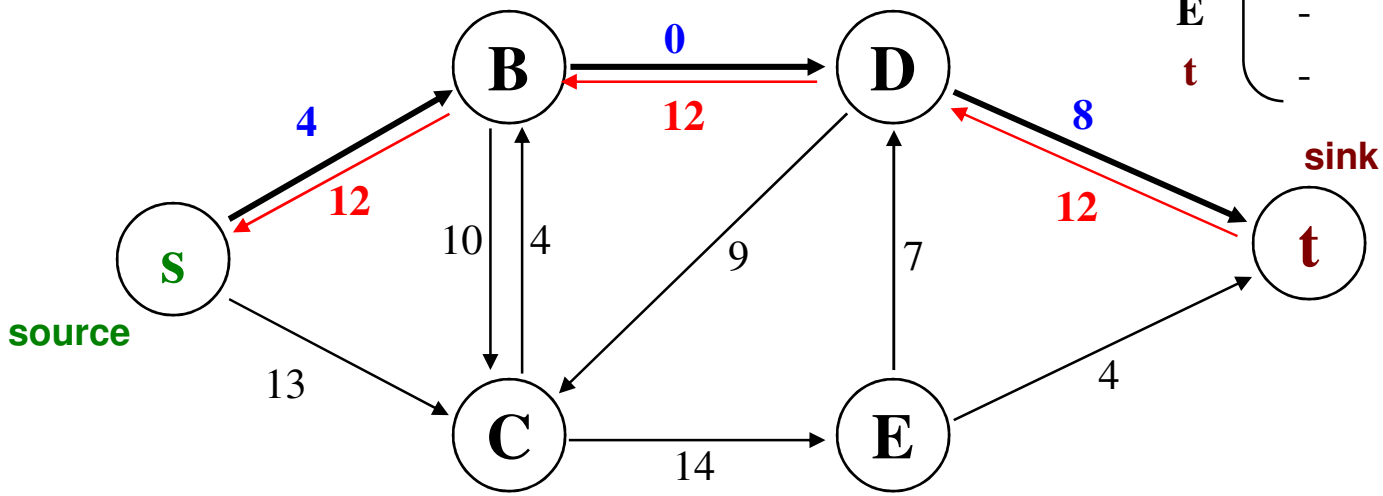
and the **red** edges? *Backwards capacities!*

	s	B	C	D	E	t
s	-	4	13	-	-	-
B	12	-	10	0	-	-
C	-	4	-	-	14	-
D	-	12	9	-	-	8
E	-	-	-	7	-	4
t	-	-	-	12	-	-

Max Flow

(Step #3) Repeat until no path exists…

	s	B	C	D	E	t
s	-	16	13	-	-	-
B	-	-	10	12	-	-
C	-	4	-	-	14	-
D	-	-	9	-	-	20
E	-	-	-	7	-	4
t	-	-	-	-	-	-



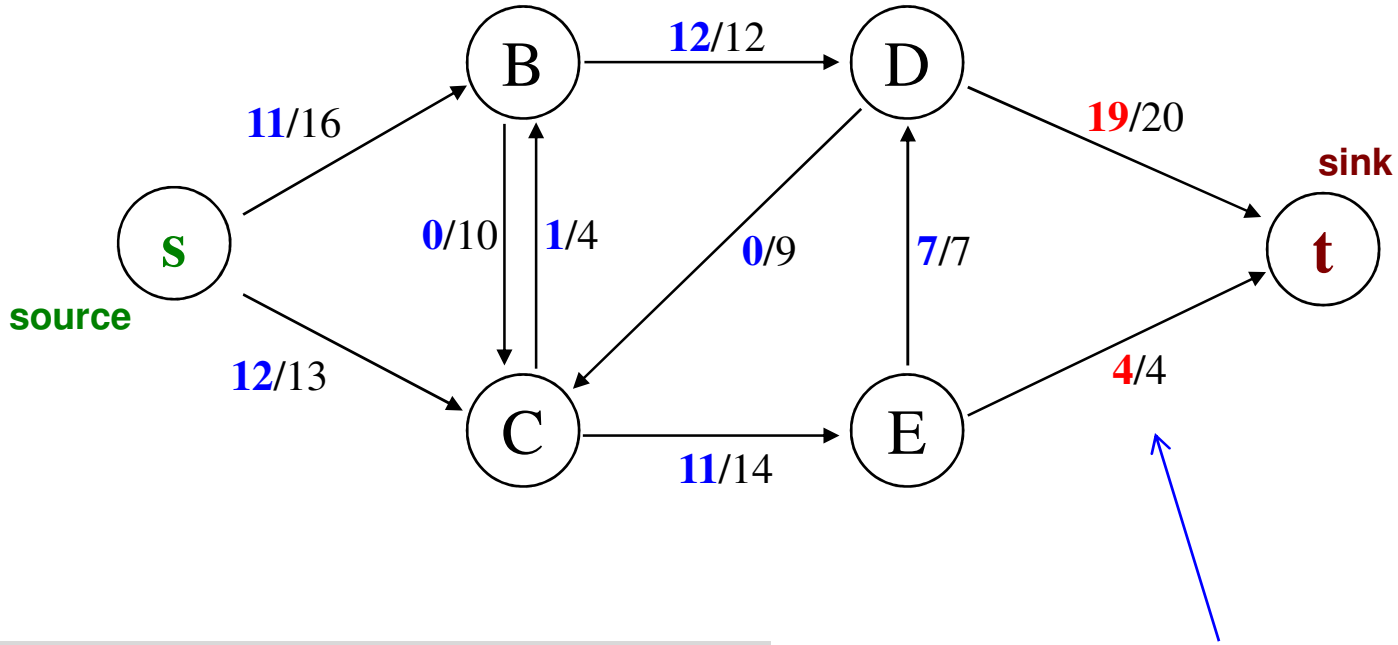
Residual capacities.

Backwards capacities.

	s	B	C	D	E	t
s	-	4	13	-	-	-
B	12	-	10	0	-	-
C	-	4	-	-	14	-
D	-	12	9	-	-	8
E	-	-	-	7	-	4
t	-	-	-	12	-	-

Max Flow

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



(Step #2) Compute RESIDUAL graph

max flow: 23

(Step #3) Repeat until no path exists…

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...

Python...

edmonds_karp algorithm

```
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!

        # find the path's flow, that is, the "bottleneck"
        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
```

Python…

```
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)):     # for each possible next node

            # 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*?

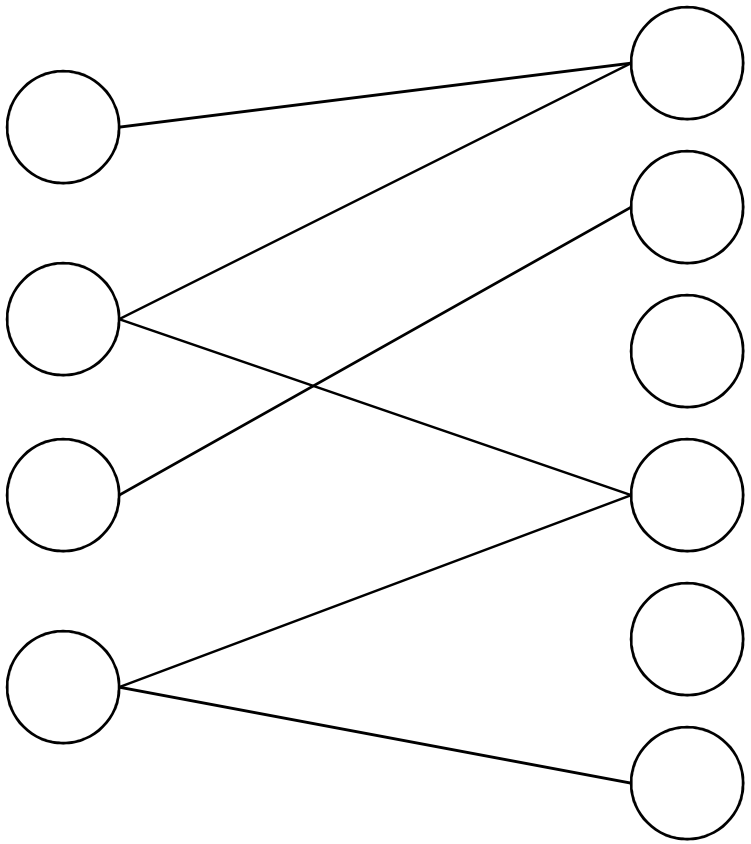
that is, beyond solving
maximum-flow problems...

Matching!

and *some* acceptable possibilities ...

we have four brides

and six grooms



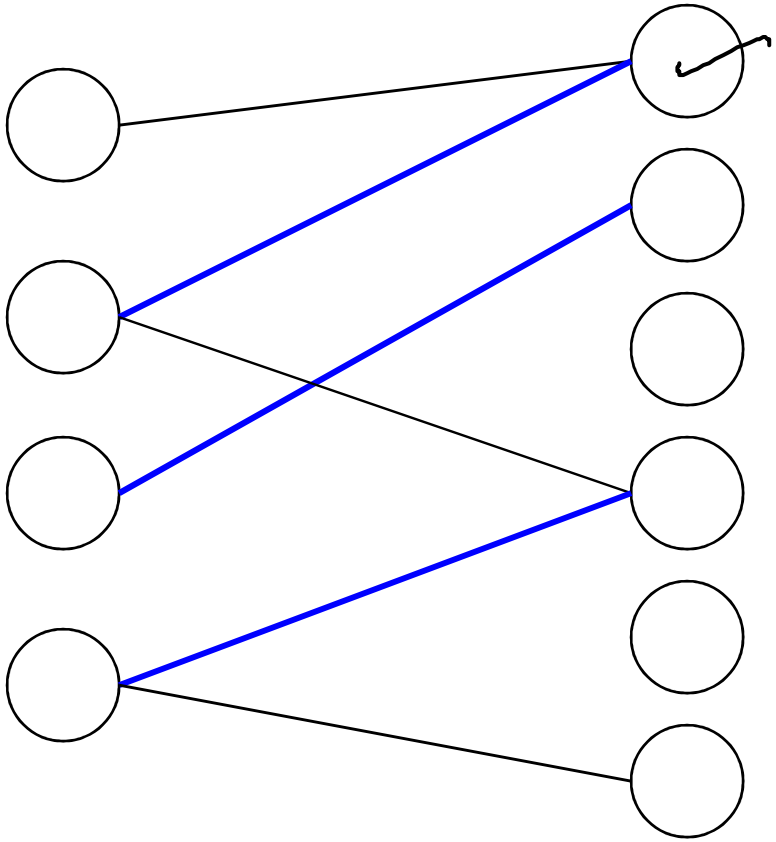
a *bipartite* graph

Matching!

and *some* acceptable possibilities ...

we have four brides

and six grooms



a ***maximal*** matching

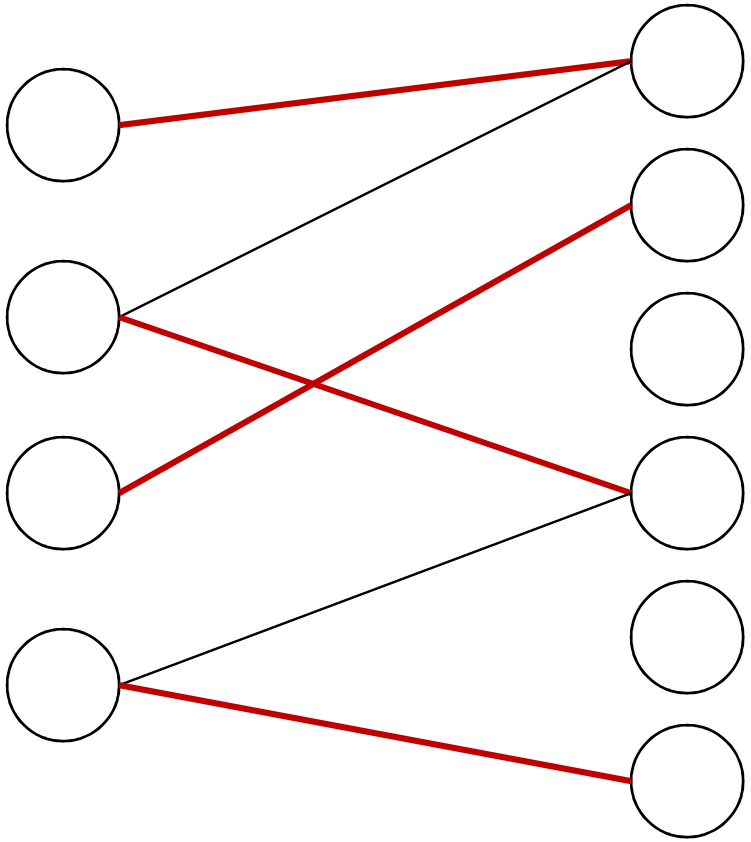
== no more matchings without rearrangement

Matching!

and *some* acceptable possibilities ...

we have four brides

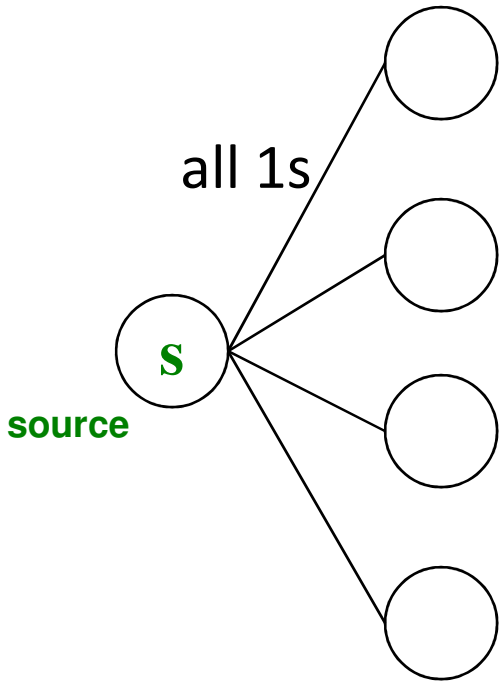
and six grooms



a ***maximum*** matching == no rearrangements will yield more matchings

Maximum matching ***is*** max flow...

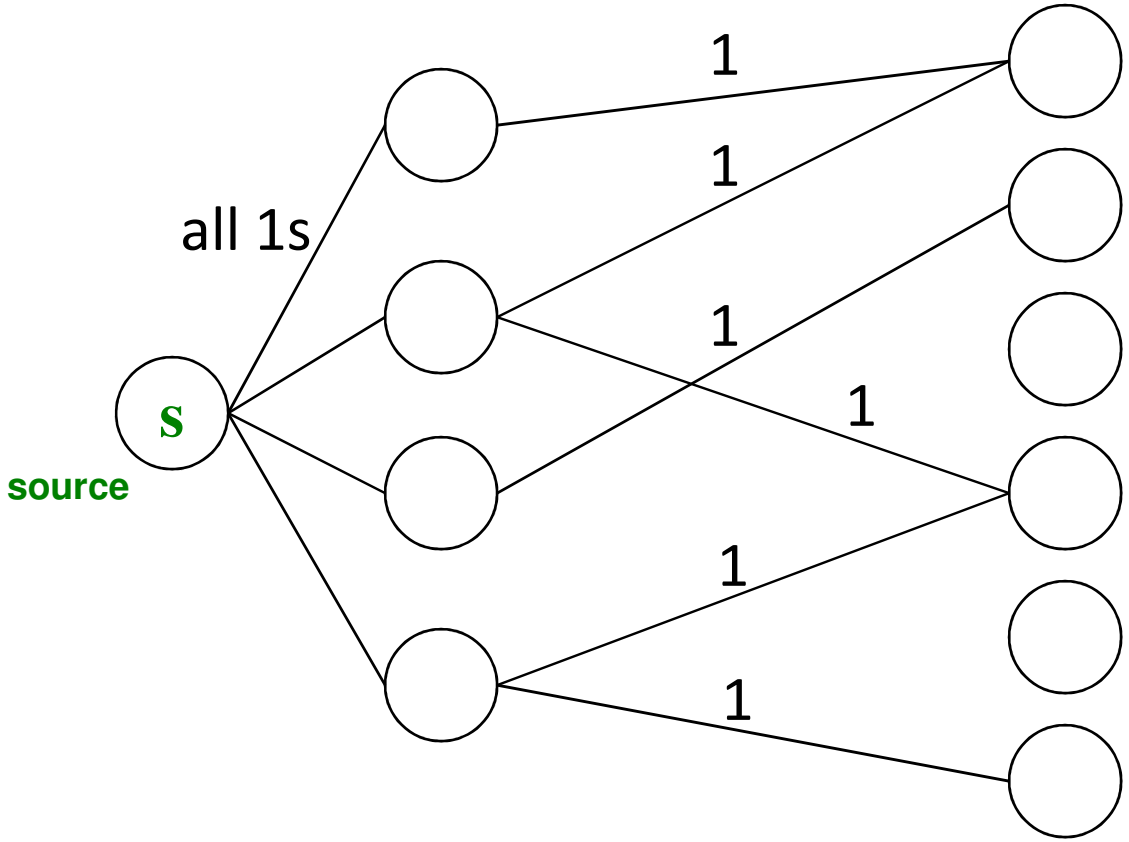
connect a source to
the left side...



Maximum matching ***is*** max flow...

connect a source to the left side...

make all capacities = 1

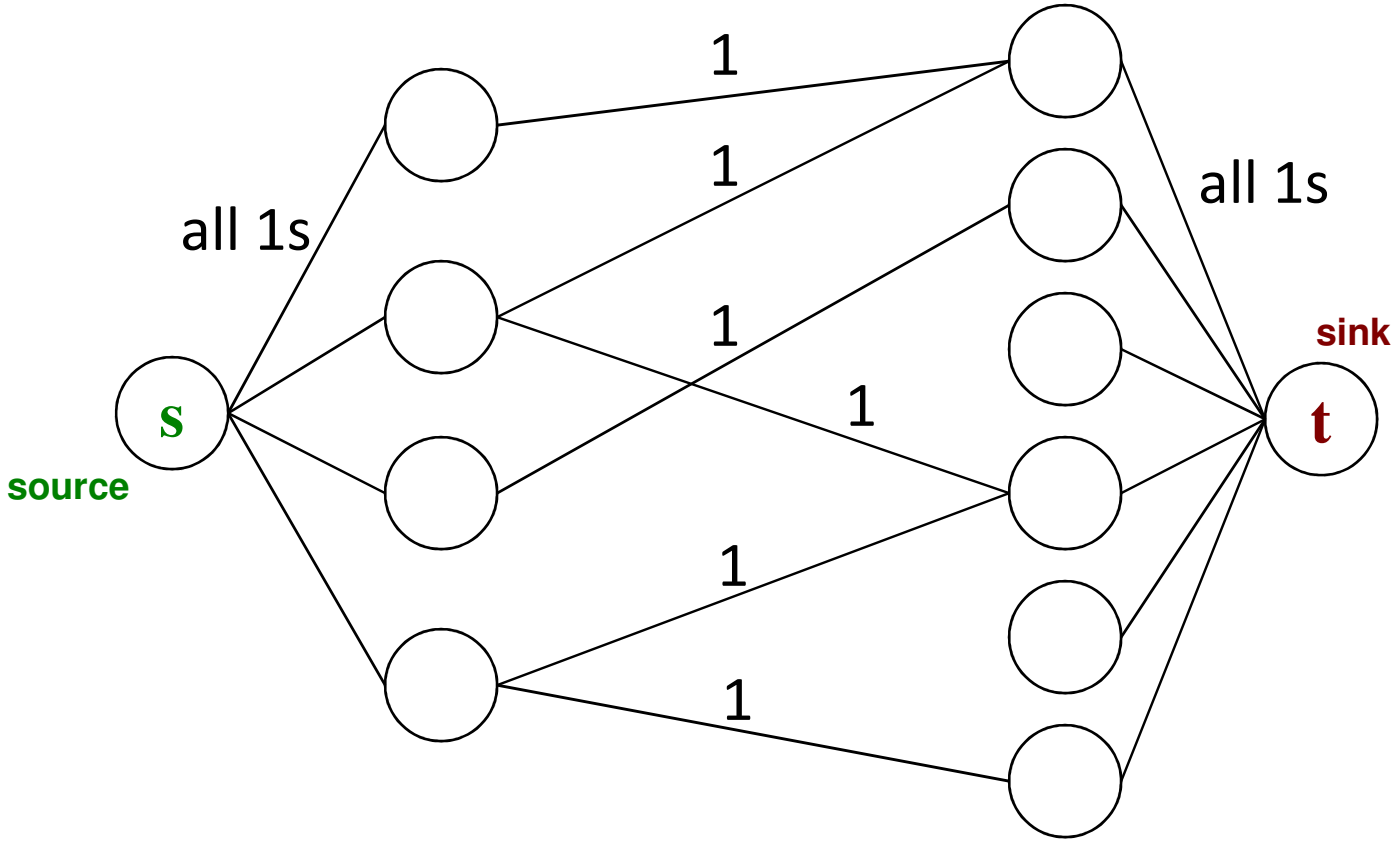


Maximum matching ***is*** max flow...

connect a source to the left side...

make all capacities = 1

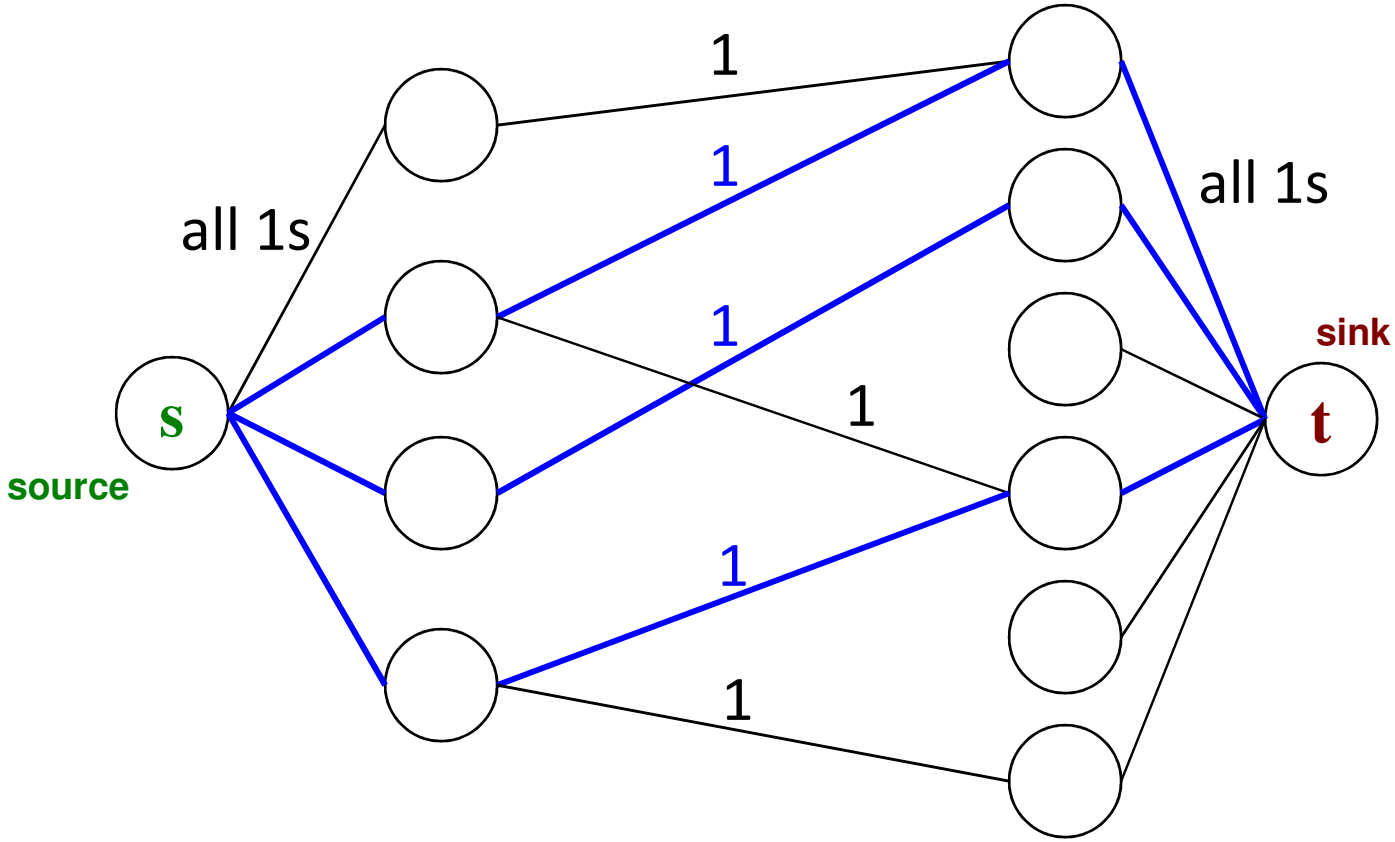
put a sink on the right



what do the source and sink constraints ensure?

Max flow thought experiment...

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

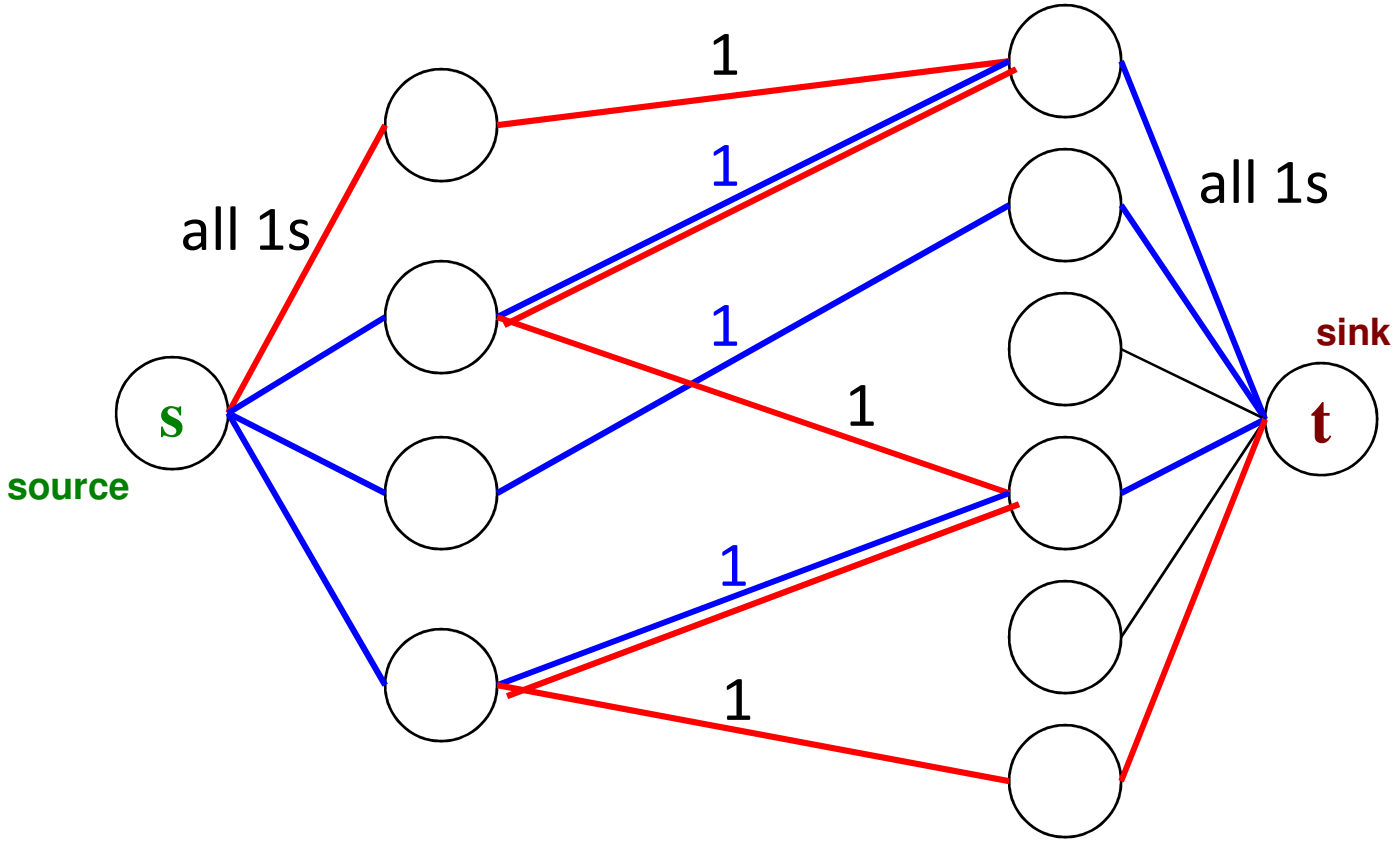


Draw what happens in the next step of the max-flow algorithm!

how to get from ***maximal*** matching to ***maximum*** matching…

Max flow thought experiment...

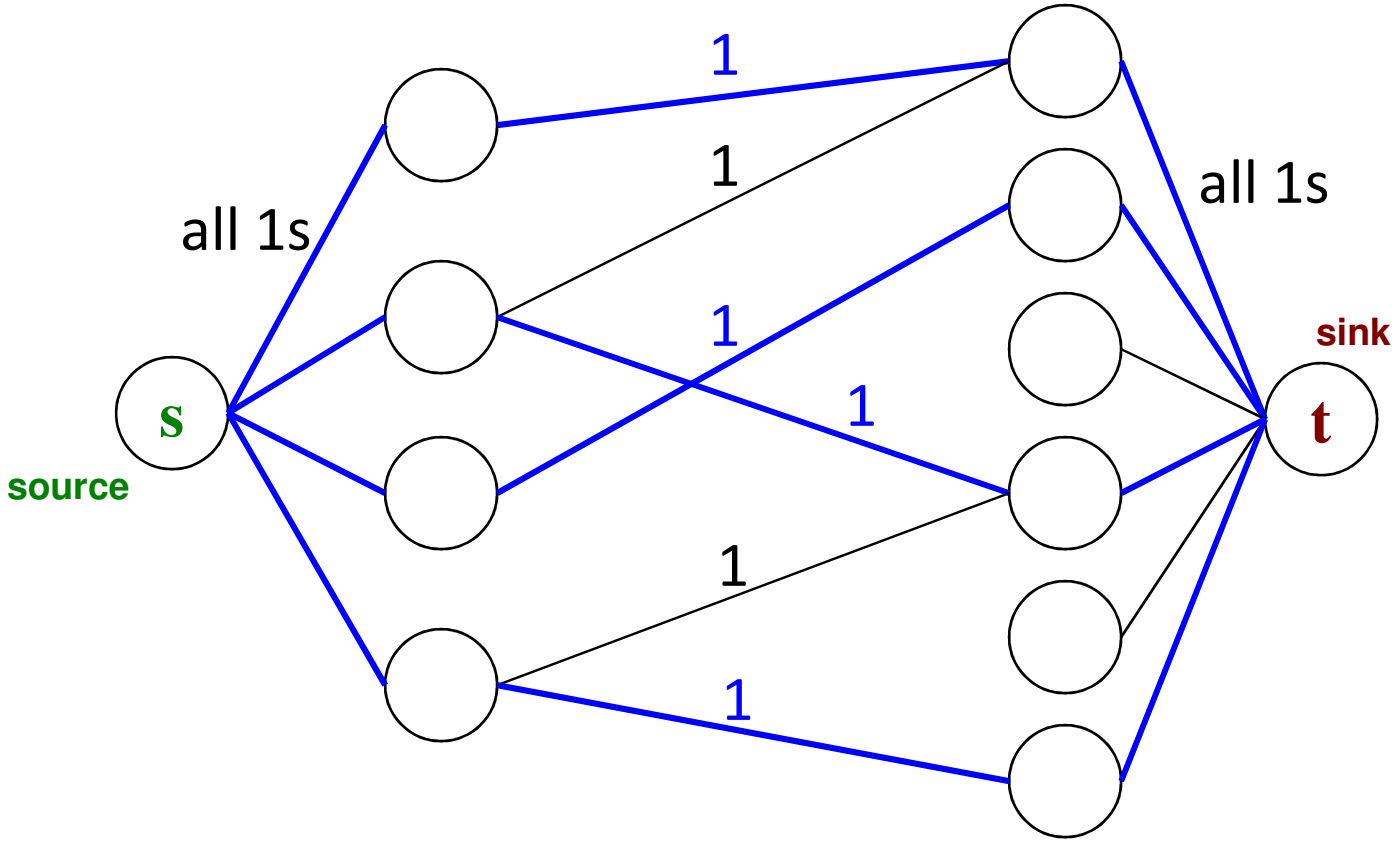
... the path it finds ...



What's going on here?

Max flow thought experiment...

Done!



Maximum matching == 4

This week's problems...

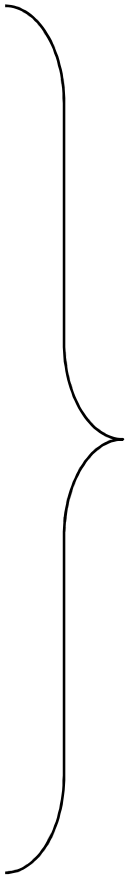
dinner

dining

hardware

muddy

feeding



all can be done
with maxflow...

The challenge:

is often ***setting up*** the graph

4 tools & 4 tasks

```
4 4
42
189
10
1000
50 1 3
20 1 3
 8 2
 3 1 4
```

hammer

phone

coffee

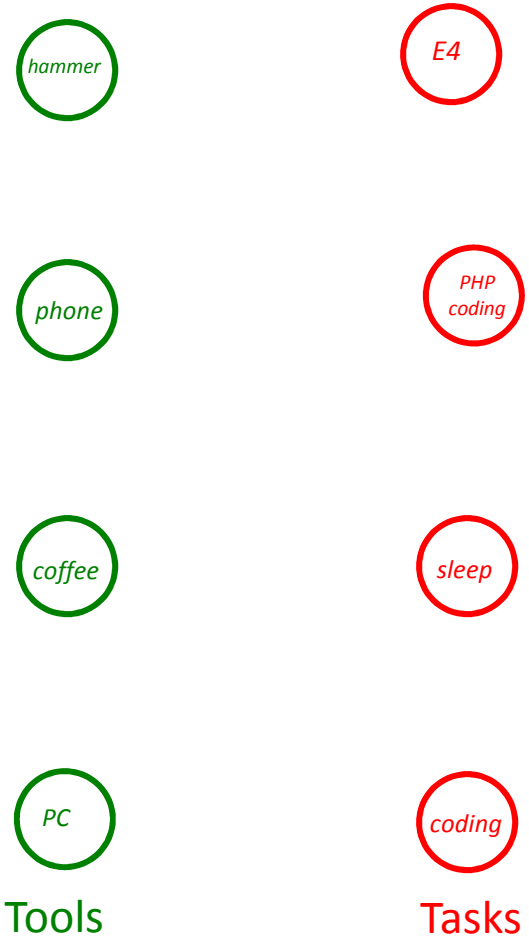
laptop

*There are four **tools** available ~ at these **costs***

*each **task** requires some **tools***

*There are four **tasks** available ~ with these **rewards***

hardware



What is flowing?

How do we use the results?

4 tools & 4 tasks

```
4 4
42
189
10
1000
50 1 3
20 1 3
 8 2
 3 1 4
```

hammer

phone

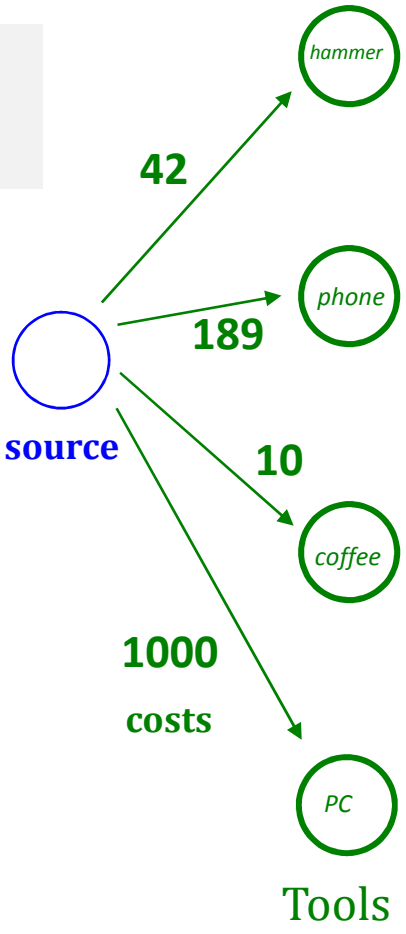
coffee

laptop

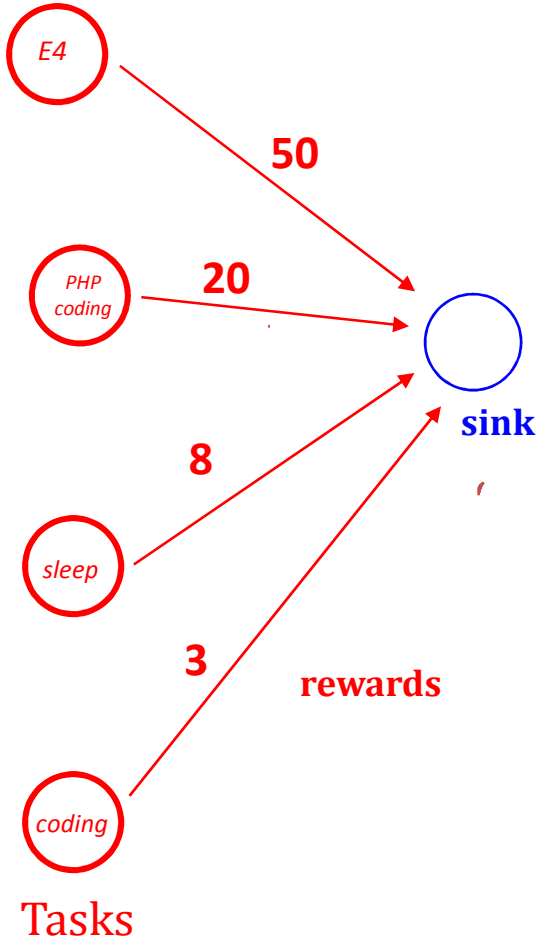
*There are four **tools** available ~ at these costs*

*each **task** requires some **tools***

*There are four **tasks** available ~ with these rewards*



hardware



What is flowing?

How do we use the results?

```
4 4
42
189
10
1000
50 1 3
20 1 3
 8 2
 3 1 4
```

4 tools & 4 tasks

hammer

phone

coffee

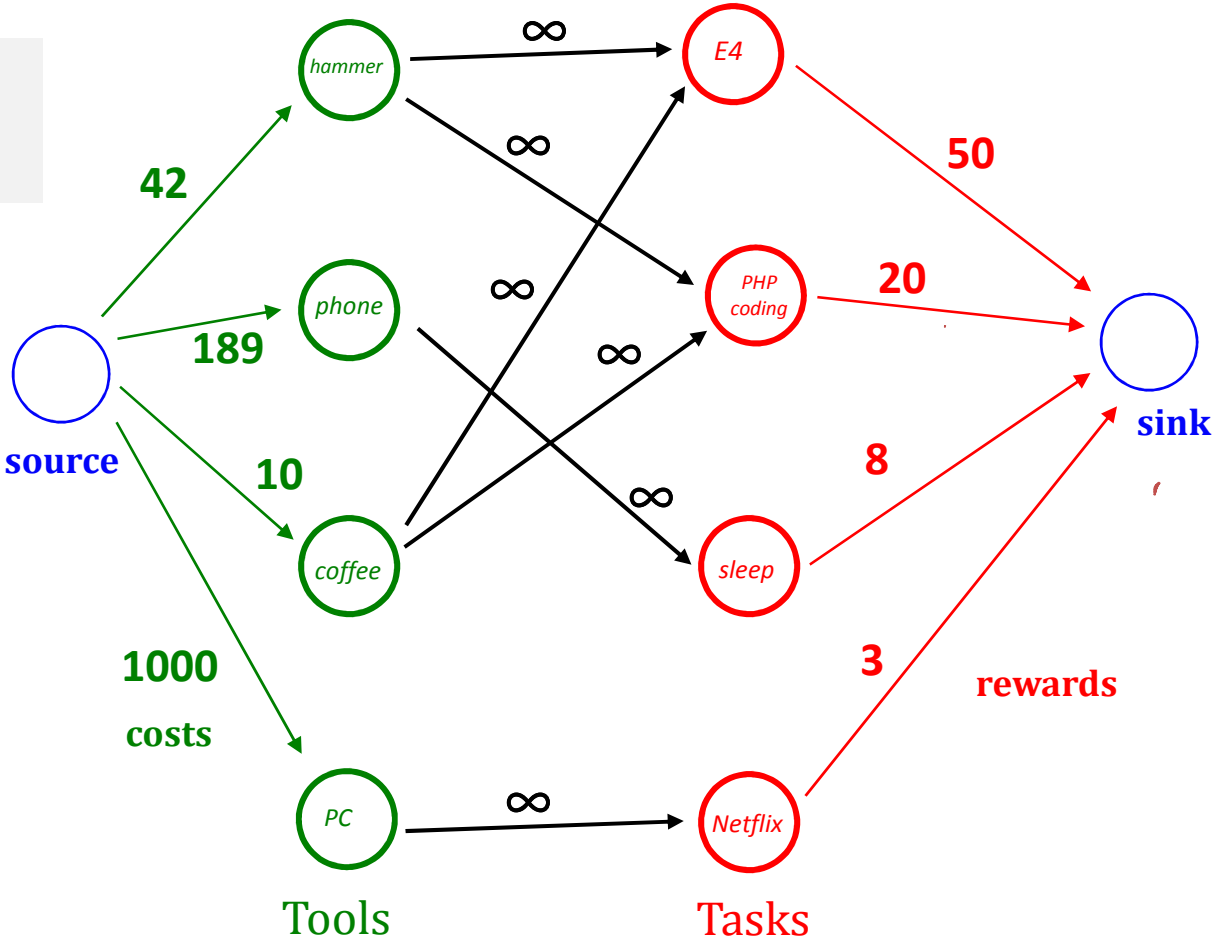
laptop

*There are four **tools** available ~ at these costs*

*each **task** requires some **tools***

*There are four **tasks** available ~ with these rewards*

hardware



What is flowing?

How do we use the results?

dinner

Input

number of teams

number of tables

# of people in each team

capacity of each table

```
4 5
4 5 3 5
3 5 2 6 4
4 5
4 5 3 5
3 5 2 6 3
0 0
```

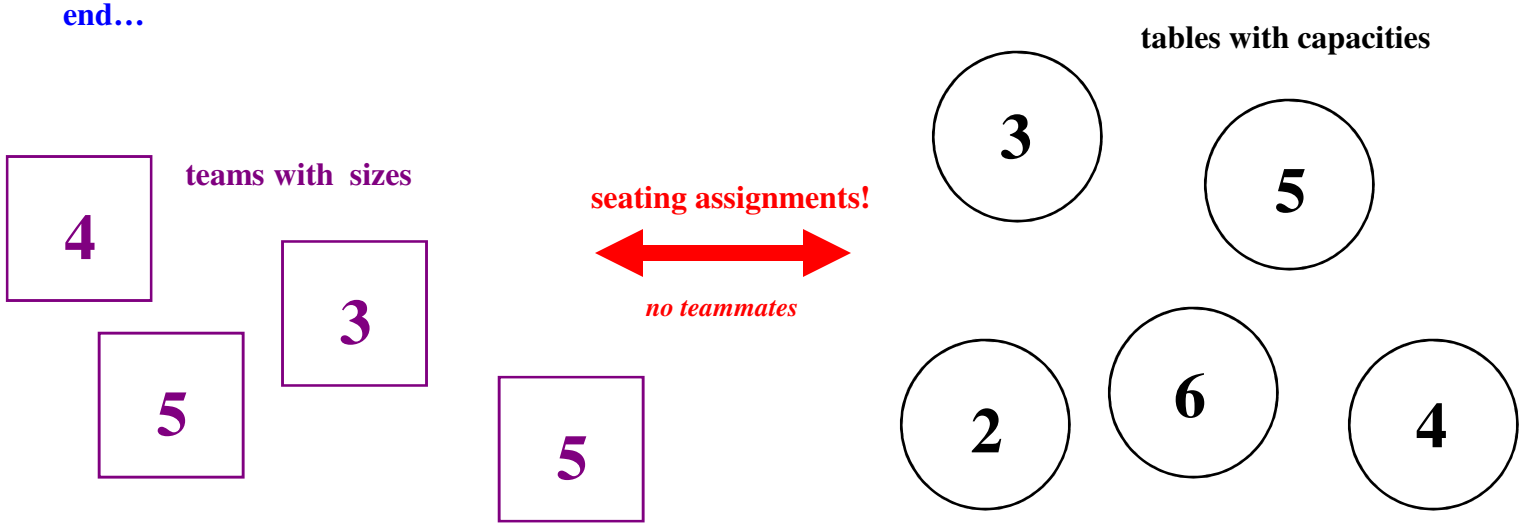
again…

end…

Output

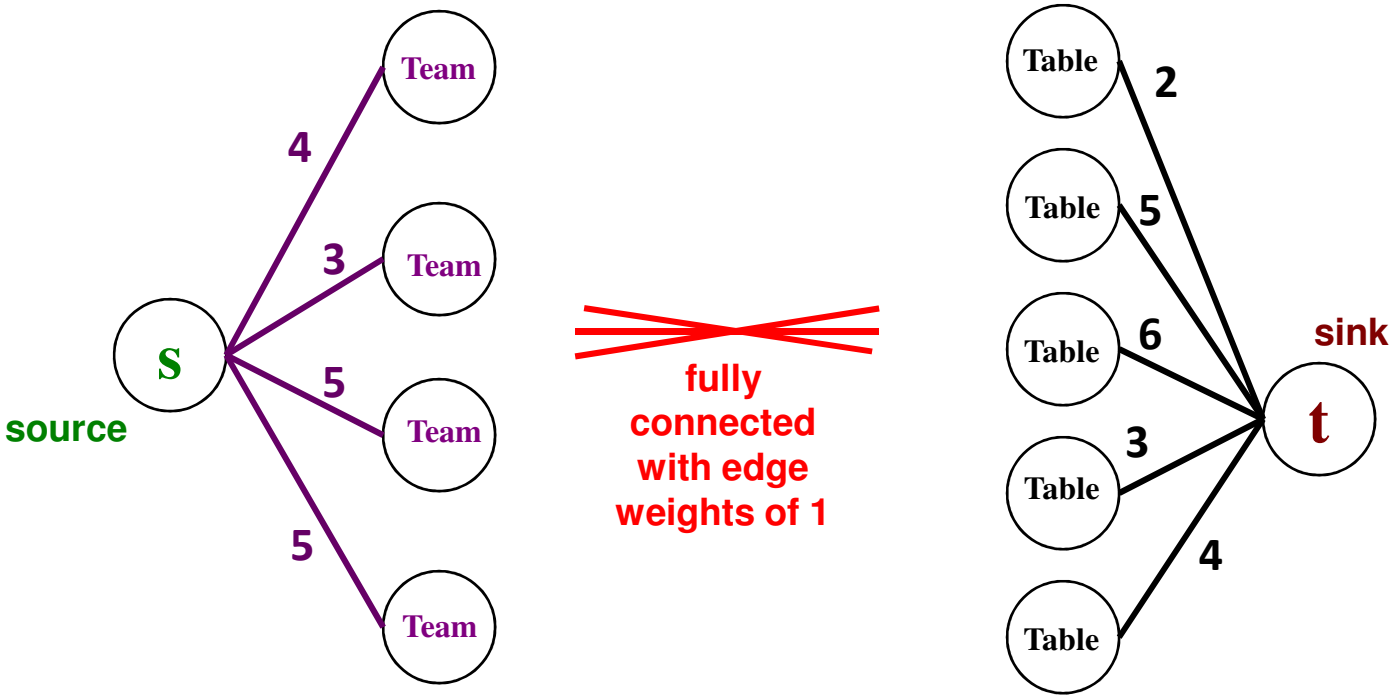
```
1
0
```

can an assignment be made without putting teammates together?



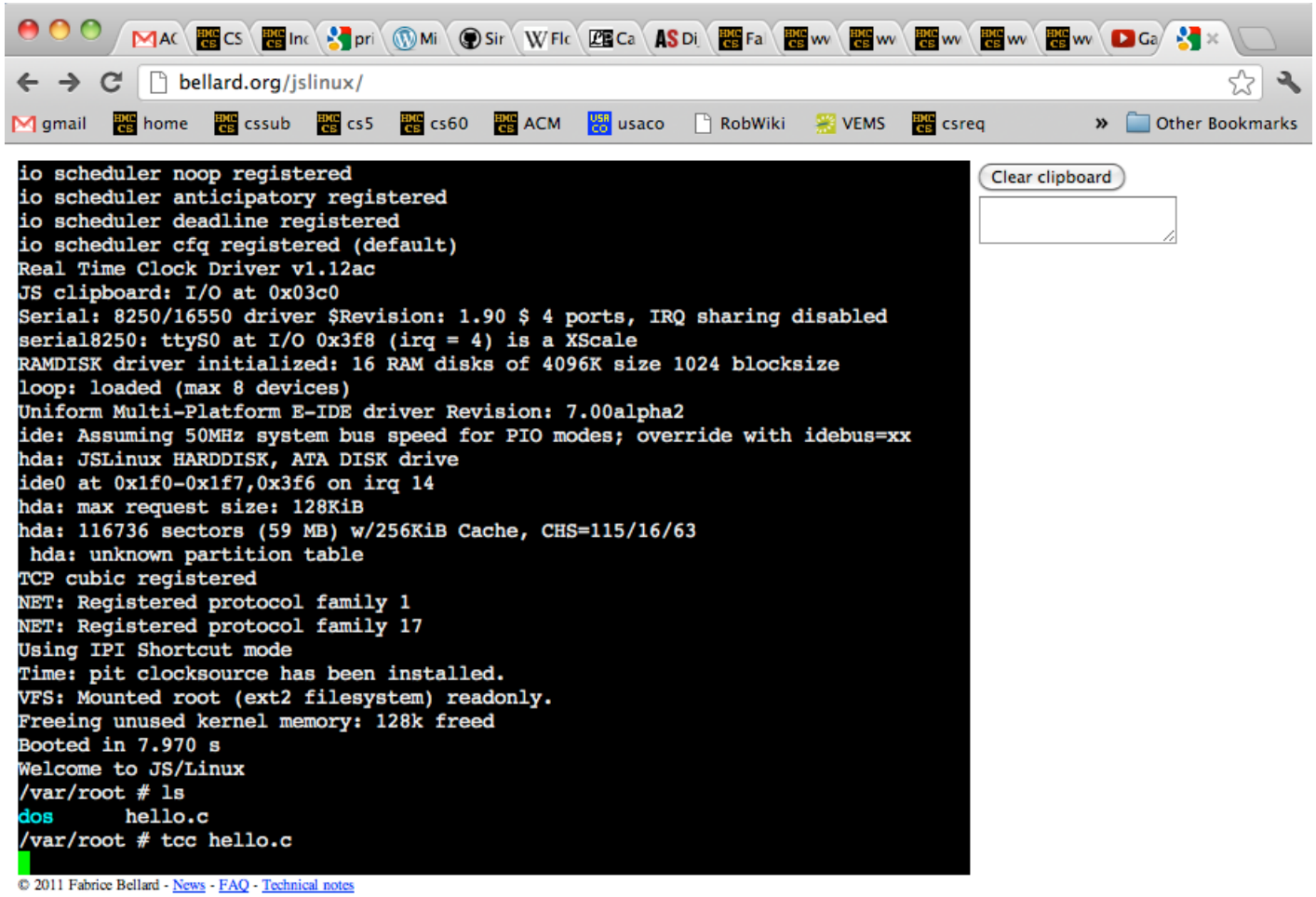
dinner

How do these edge weights reflect the problem constraints?



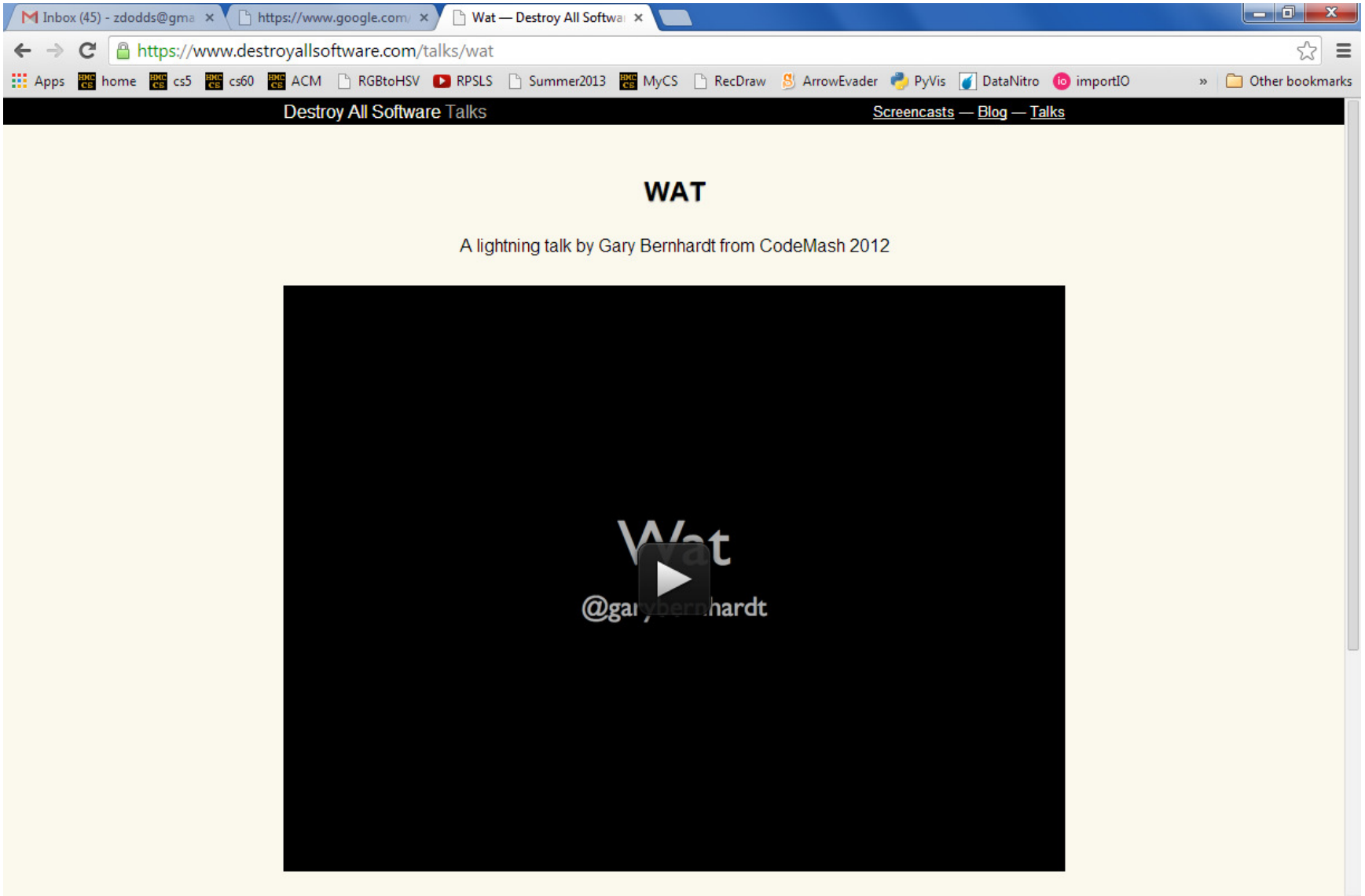
How does maxflow help?

What?!



Oddities from computer code…

original *Wat* talk...



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

PHP's
WAT!

PHP's *WAT!*

```
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";
```

Python's *WAT!*

WAT?

```
1 == True
```

```
0 == False
```

Are these

True

or

False

?

```
(2==2)==2
```

```
(1==1)==1
```

```
0==(0==0)
```

```
0==(1==0)
```

```
1==1==1
```

```
0==0==0
```

Good luck through April!

Jotto guess!

April is the cruellest month…

T.S. Eliot

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 -

	diner	bloat	gumbo
FYears (Dan/Sam):	2	0	0
SophSelectTeam:	1	1	0
SophFunTeam:	1	3	0
JRs:	1	1	0
SRs:	0	0	0
ExtraMudd:	1	1	2
"Other" (me):	1	1	0

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...