ORF 363/COS 323

Computing and Optimization in the Physical and Social Sciences

Amir Ali Ahmadi
Princeton, ORFE

Lecture 1

What is optimization?

- Roughly, can think of optimization as the science of making the most out of every situation.
- You've probably all done it many times recently:

- What courses to take?
  - To maximize learning.
  - To maximize GPA (?!)
  - Courses can't conflict.
  - Not before 10AM.
  - Professor rating > 4.5.

- What furniture to buy?
  - To minimize cost.
  - To maximize comfort.
  - Must fit in your room.
  - Must have 3 drawers.
  - Not too heavy.

- What path to choose for a run?
  - Minimize (2020 version) or maximize (2019 version) encounter with other runners.
  - Path must be between 5 and 7 miles.
  - Chosen roads must have side walks.
  - Path should get you back home.

- **Common theme:**
  - You make decisions and choose one of many alternatives.
  - You hope to maximize or minimize something (you have an objective).
  - You cannot make arbitrary decisions. Life puts constraints on you.

How is this class different from your every-day optimization?

- We'll be learning techniques for dealing with problems that have
  - Thousands (if not millions) of variables
  - Thousands (if not millions) of constraints
- These problems appear every day in the industry, in science, in engineering
- Hopeless to make decisions in your head and with rules of thumb
- Need mathematical techniques that translate to algorithms
  - Algorithms then get implemented on a computer to solve your optimization problem
- We typically model a physical or social scenario with a precise mathematical description
- In this mathematical model, we care about actually finding *the best solution*
- Whenever we can't find the best solution, we would like to know how far off our proposed solution is

Examples of optimization problems

In finance

- In what proportions to invest in 500 stocks?
  - To maximize return.
  - To minimize risk.

  - No more than 1/5 of your money in any one stock.
  - Transactions costs < $70.
  - Return rate > 2%.

In control engineering

- How to drive an autonomous vehicle from A to B?
  - To minimize fuel consumption.
  - To minimize travel time.

  - Distance to closest obstacle > 2 meters.
  - Speed < 40 miles/hr.
  - Path needs to be smooth (no sudden changes in direction).

In machine learning

- How to assign likelihoods to emails being spam?
  - To minimize probability of a false positive.
  - To penalize overfitting on training set.

  - Probability of false negative < .15.
  - Misclassification error on training set < 5%.

Computing and Optimization

- This class will give you a broad introduction to

  "optimization from a computational viewpoint."

- Optimization and computing are very close areas of applied mathematics:
  - For a host of major problems in computer science, the best algorithms currently come from the theory of optimization.
  - Conversely, foundational work by computer scientists has led to a shift of focus in optimization theory from "mathematical analysis" to "computational mathematics."

- Several basic topics in scientific computing (that we'll cover in this course) are either special cases or fundamental ingredients of more elaborate optimization algorithms:
  - Least squares, root finding, solving linear systems, solving linear inequalities, approximation and fitting, matrix factorizations, conjugate gradients,…

Agenda for today

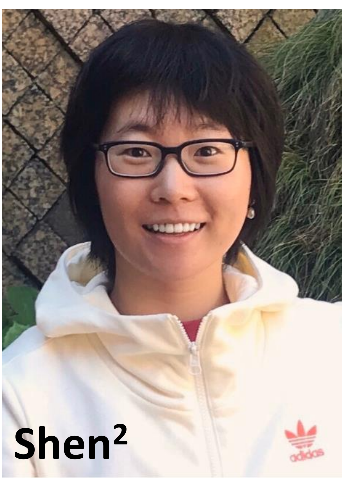
- Meet your teaching staff
- Get your hands dirty with algorithms
  - Game 1
  - Game 2
- Modelling with a mathematical program
  - Fermat's last theorem!
- Course logistics and expectations

Meet your teaching staff (1/3)

- **Amir Ali Ahmadi** (Amir Ali, or Amirali, is my first name)

http://aaa.princeton.edu/ aaa@p...

- I am a Professor at ORFE, and affiliated faculty at COS, EE, MAE, PACM, CSML
- I came to Princeton from MIT, EECS, after a fellowship at IBM Research
- Office hours: **Wednesdays, 3-4:30pm**
(all office hours and precepts have the same Zoom link – all times are EST)



- **Shen Shen** (co-instructor and preceptor)

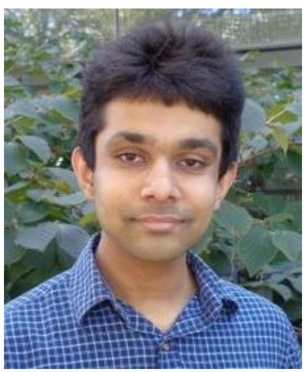
https://shensquared.github.io/ shenshen@mit.edu

- Postdoc and Lecturer at ORFE
- Joins us from MIT, EECS (Robot Locomotion Group)
- Office hours: **Tuesdays, 4:30-6pm & Wednesdays 4:30-6pm**

Meet your teaching staff (2/3)

- **Grace Lee (AI)**
- Office hours: **Mon 9-10:30am**
- gylee@p...

- **Sobhan Miryoosefi (AI)**
- Office hours: **Mon 4:30-6pm**
- miryoosefi@p...

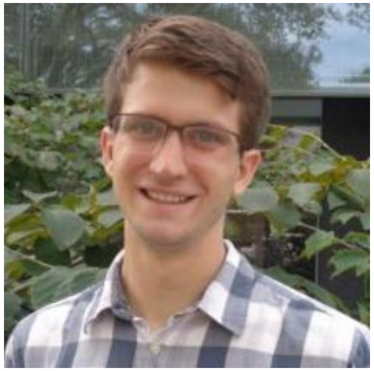


- **Abraar Chaudhry (1/2 AI)**
- Office hours: **Mon 7-8:30pm**
- azc@p...

- **Xiaohe Luo (AI)**
- Office hours: **Tue 9-10:30am**
- xiaohel@p...

Meet your teaching staff (3/3)

- **Brian Cheung (AI)**
- Office hours: **Tue 7-8:30pm**
- bcheung@p...



- **William Underwood (AI)**
- Office hours: **Wed 7-8:30pm**
- wgu2@p...

- **Cemil Dibek (honorary AI)**
- Office hours: **Thu 9-10:30am**
- cdibek@p...

Course website, Office Hours, and Precepts

Course website:
http://aaa.princeton.edu/orf363

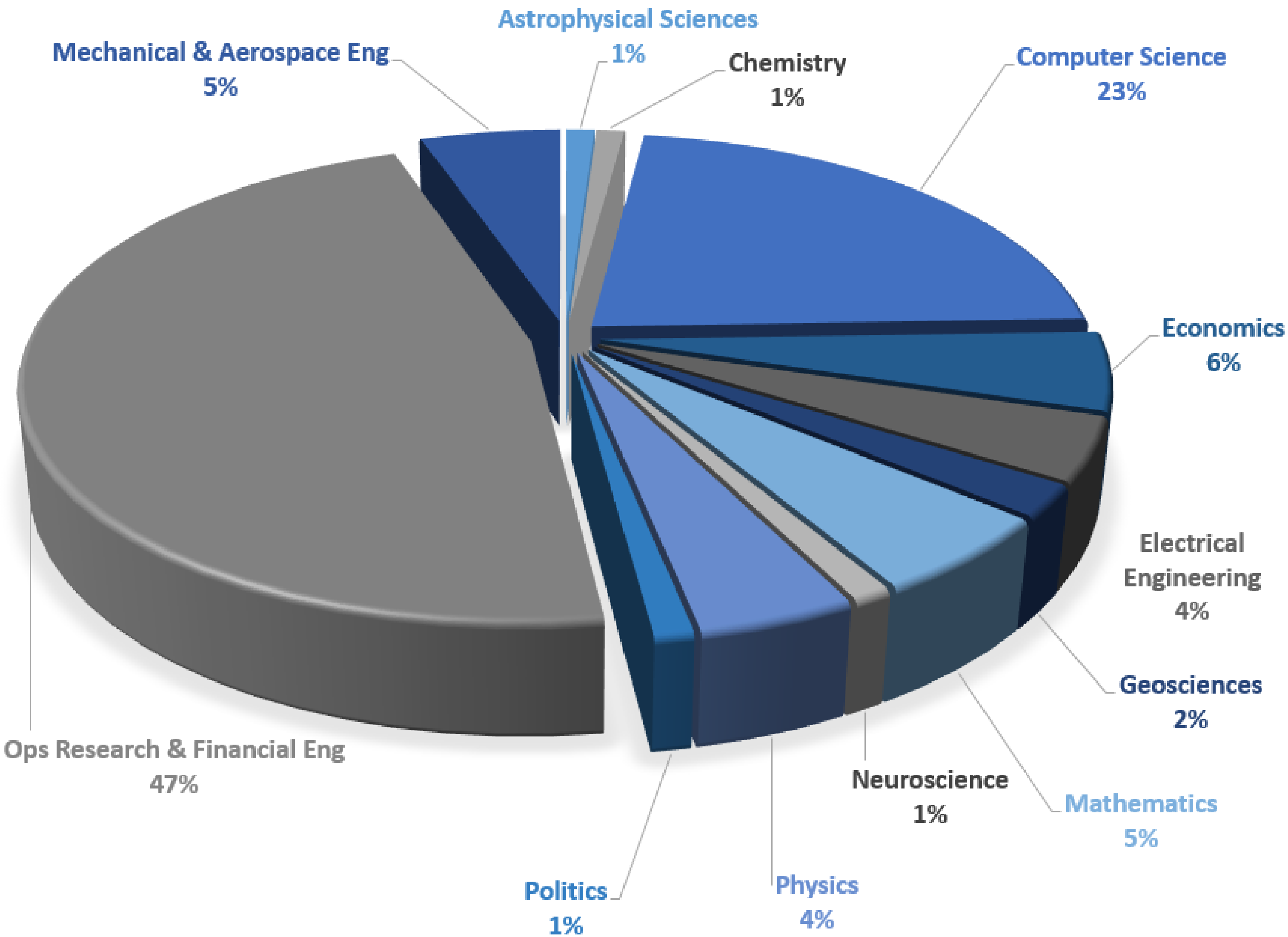
Zoom links on course website and on Blackboard (password has been emailed to you).
Office hours and precepts start right away!

- **10 weekly office hours:**
  - **Mondays: 9-10:30am (Grace), 4:30-6pm (Sobhan), 7-8:30pm (Abraar)**
  - **Tuesdays: 9-10:30am (Xiaohe), 4:30-6pm (Shen), 7-8:30pm (Brian)**
  - **Wednesdays: 3-4:30pm (AAA), 4:30-6pm (Shen), 7-8:30pm (William)**
  - **Thursdays: 9-10:30am (Cemil)**
- **6 weekly precepts (each with ~15 students):**
  - **Thursdays: 4:30-5:20pm, 6-6:50pm**
  - **Friday mornings: 9-9:50am, 11-11:50am**
  - **Friday afternoons: 1:30-2:20pm, 3:30-4:20pm**
- **Piazza:**

Sign up at http://piazza.com/princeton/fall2020/orf363
(password has been emailed to you)

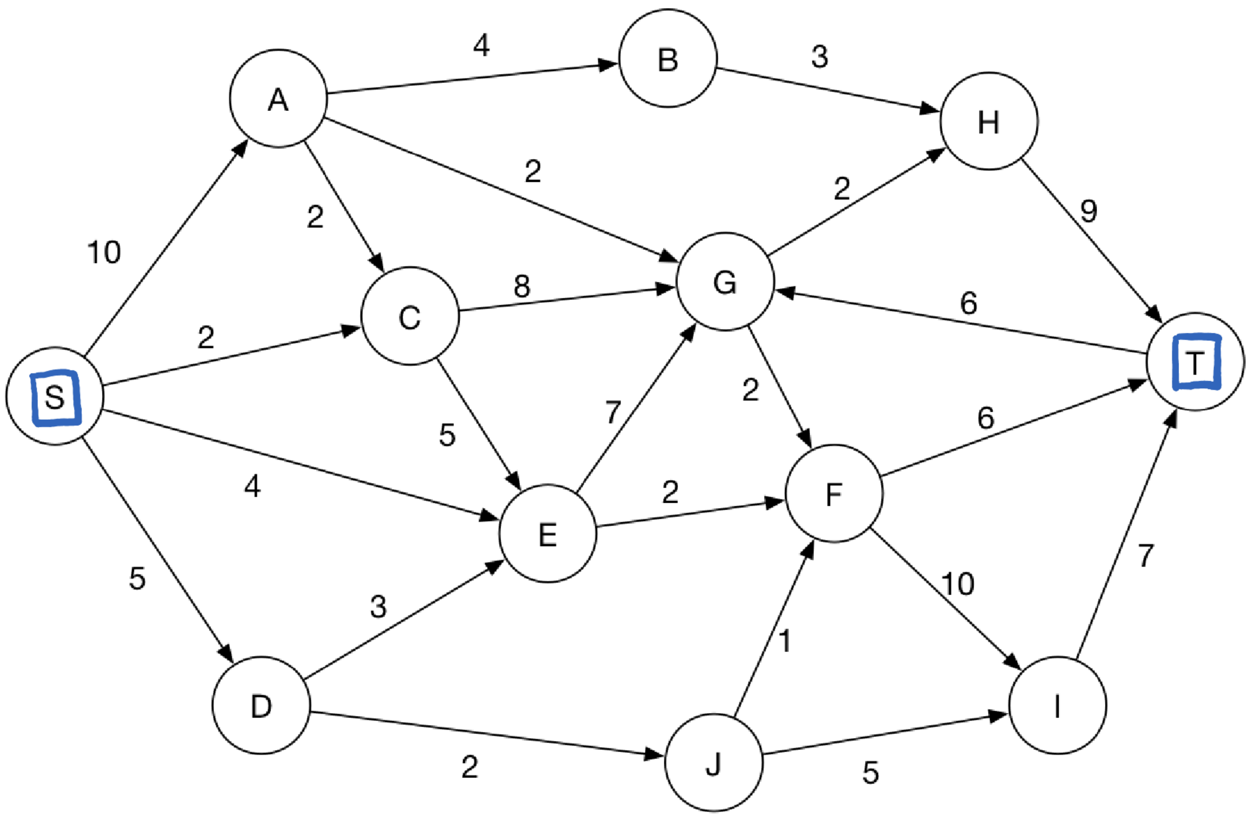
Meet your classmates

ORF 363/COS 323, Fall 2020 (94 students)



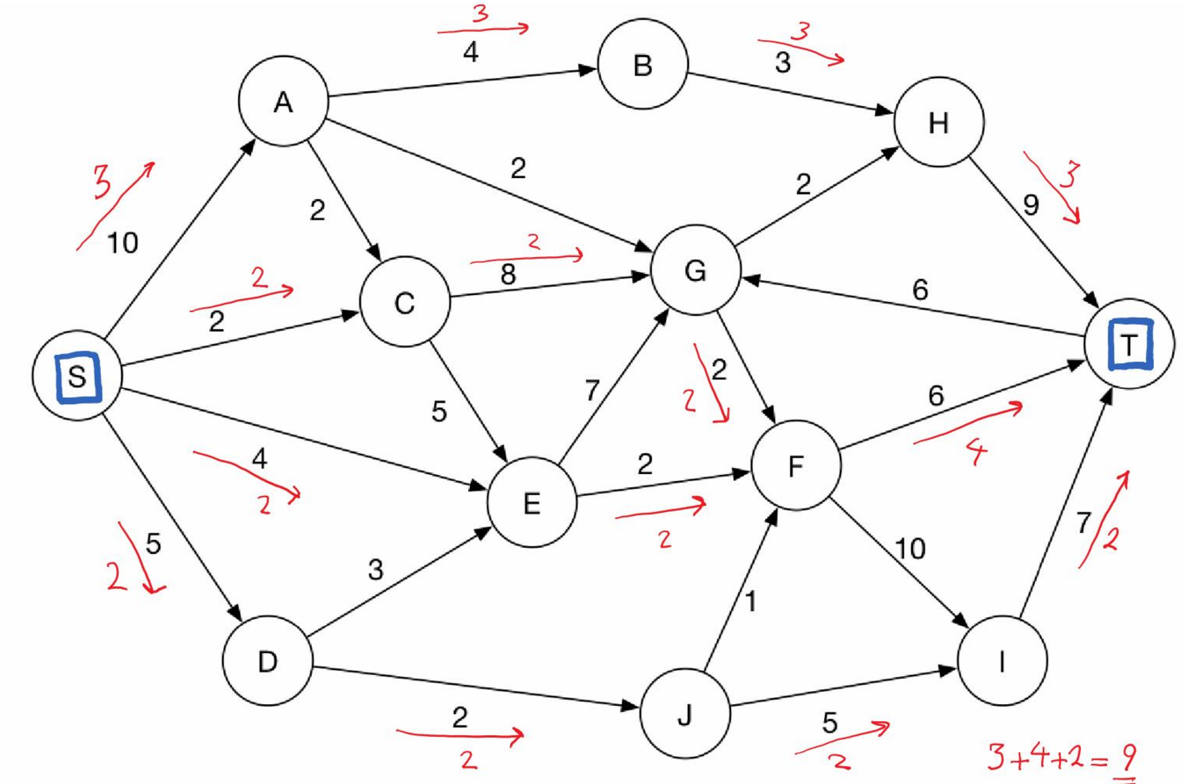
Let's get to the games!

Let’s ship some oil together!



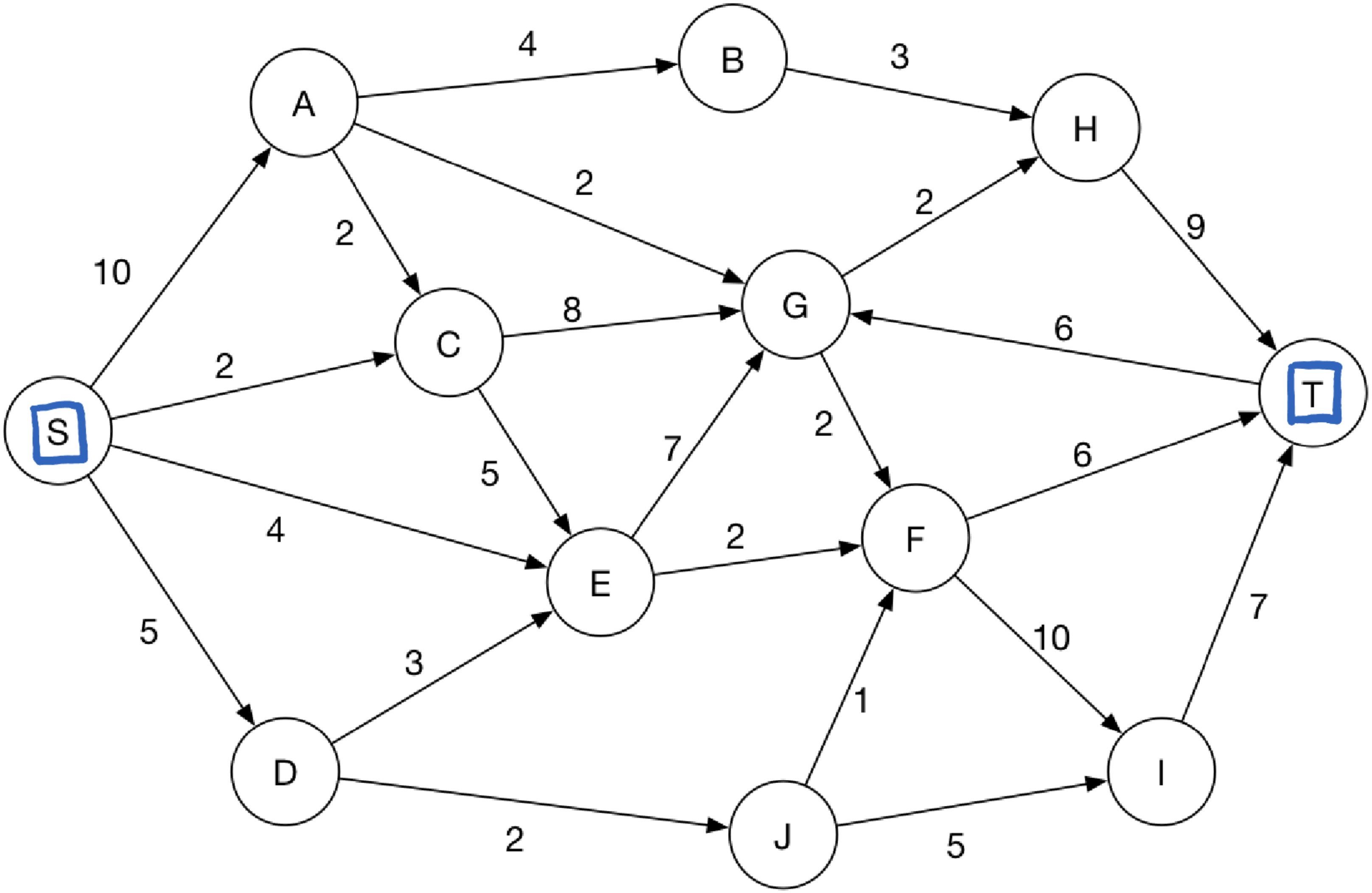
- **Rules of the game:**
  - Cannot exceed capacity on the edges.
  - For each node, except for S ant T, flow in = flow out (i.e., no storage).
  - **Goal:** ship as much oil as you can from S to T.

- Let me start things off for you. Here is a flow with value 9:

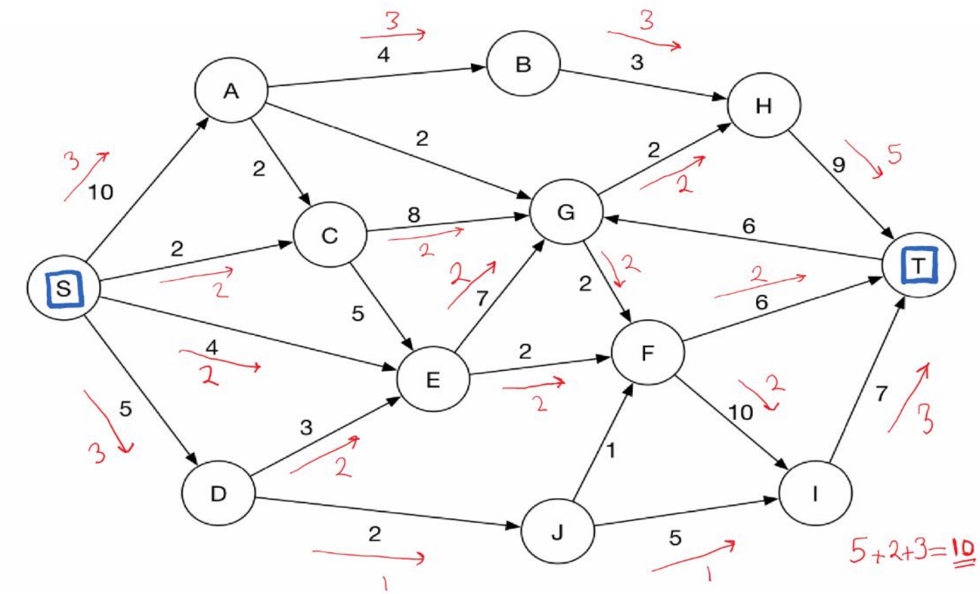


- Can you do better? How much better?
- You all get a copy of this graph.
  - You have 6 minutes!

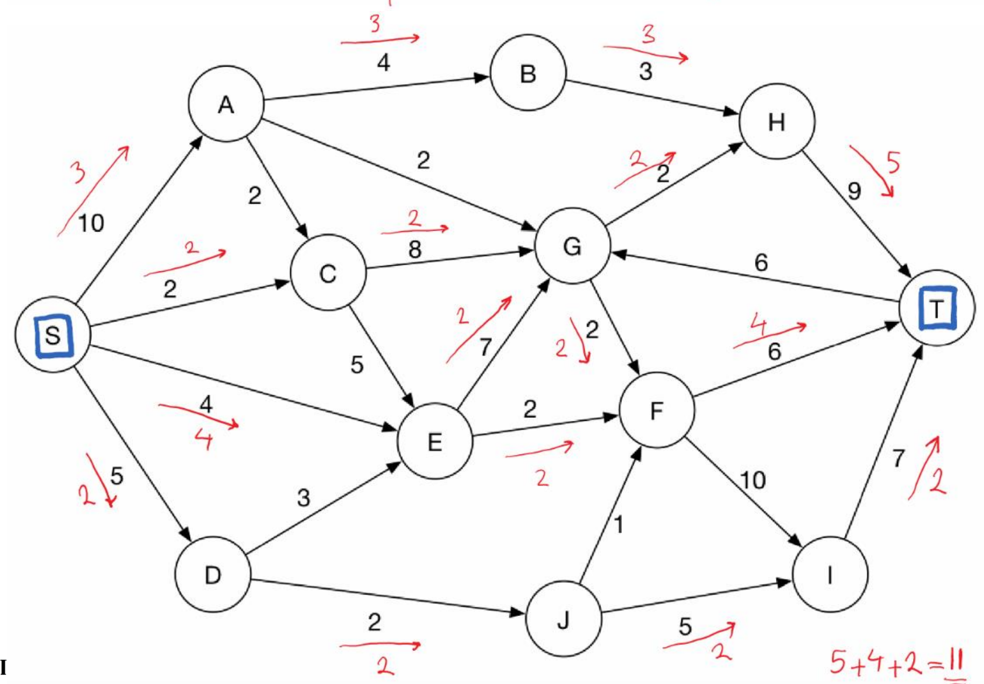
You tell me, I draw…



A couple of good attempts



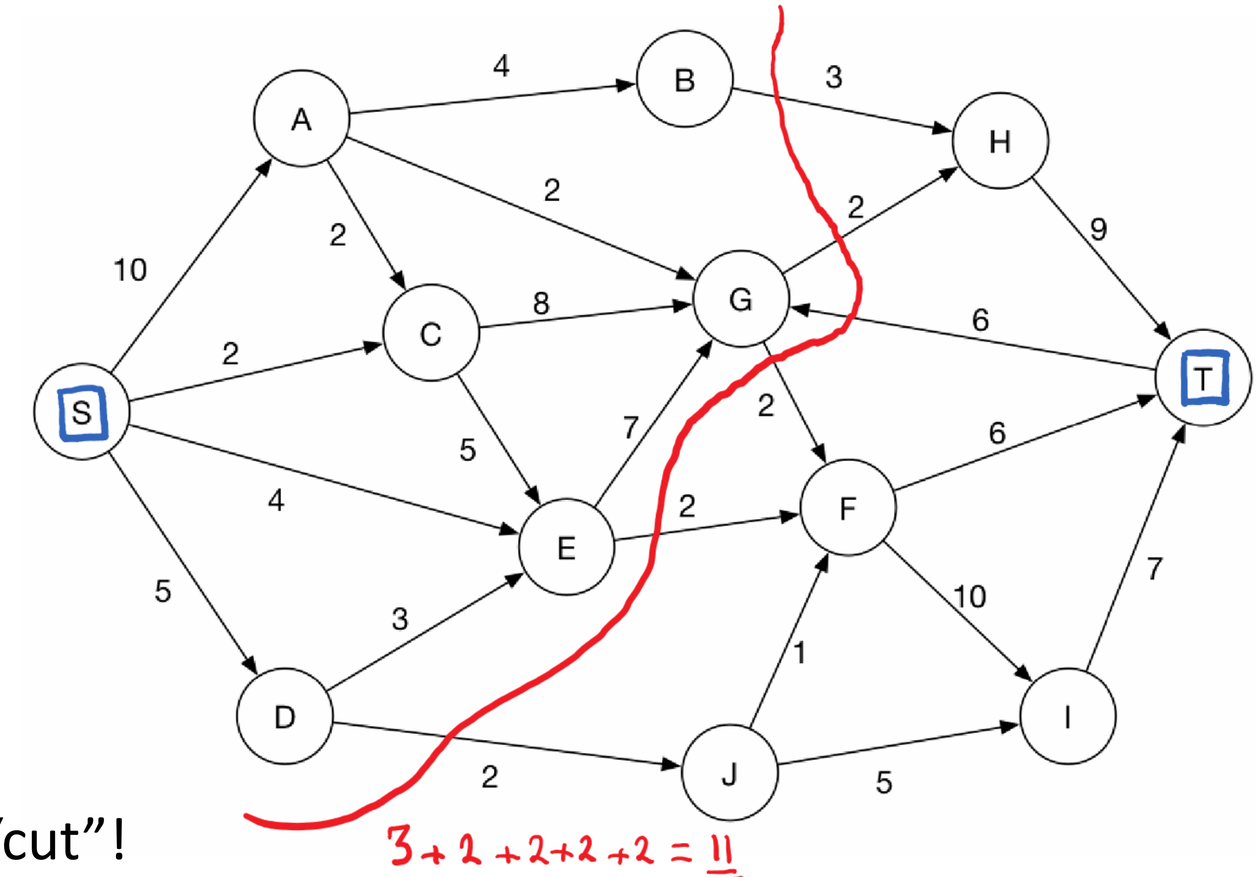
- Flow of value 10
- Can you do better?



- Flow of value 11
- Can you do better?
- How can you prove that it's impossible to do better?

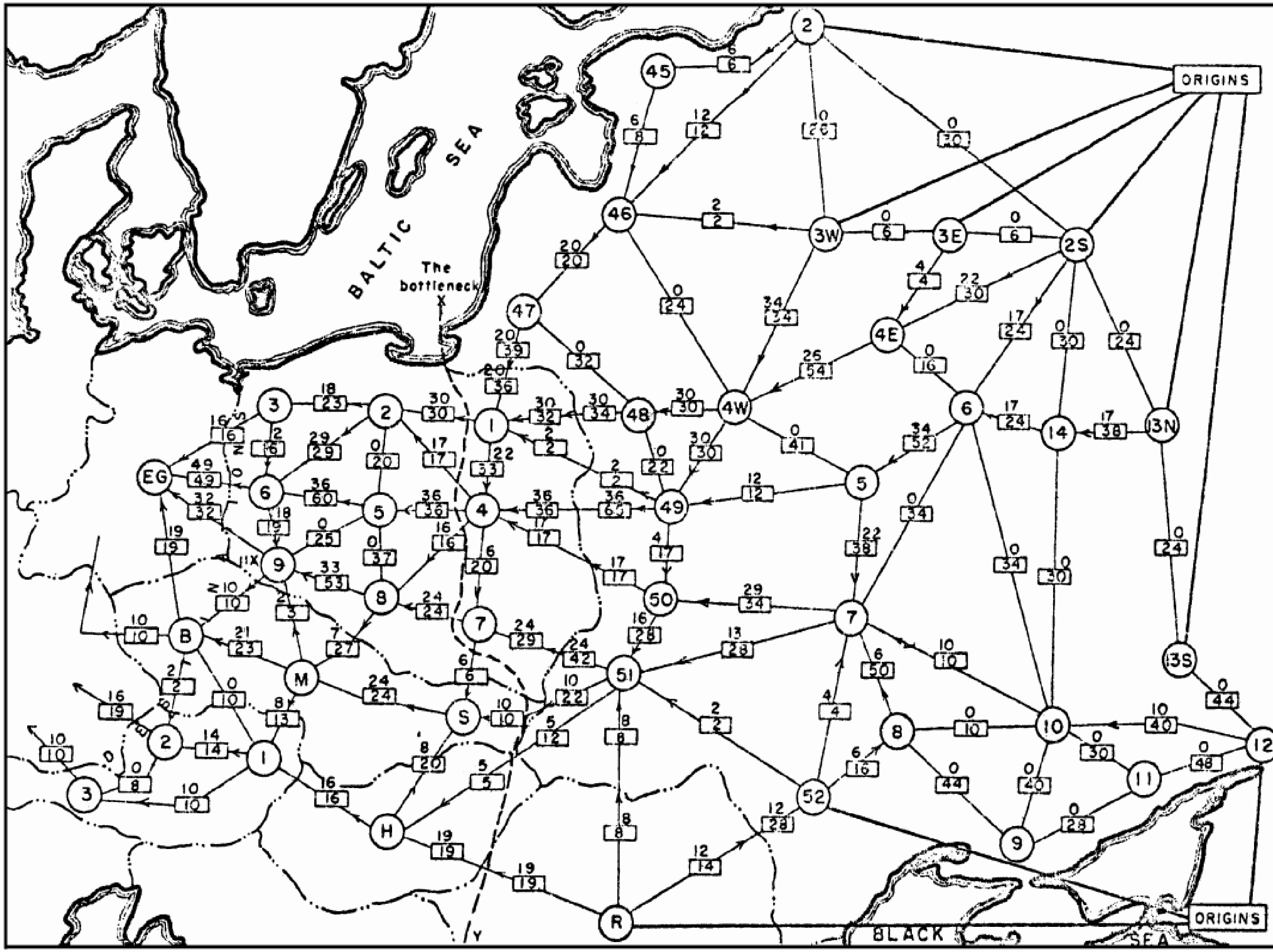
11 is the best possible!

- Proof by magic:



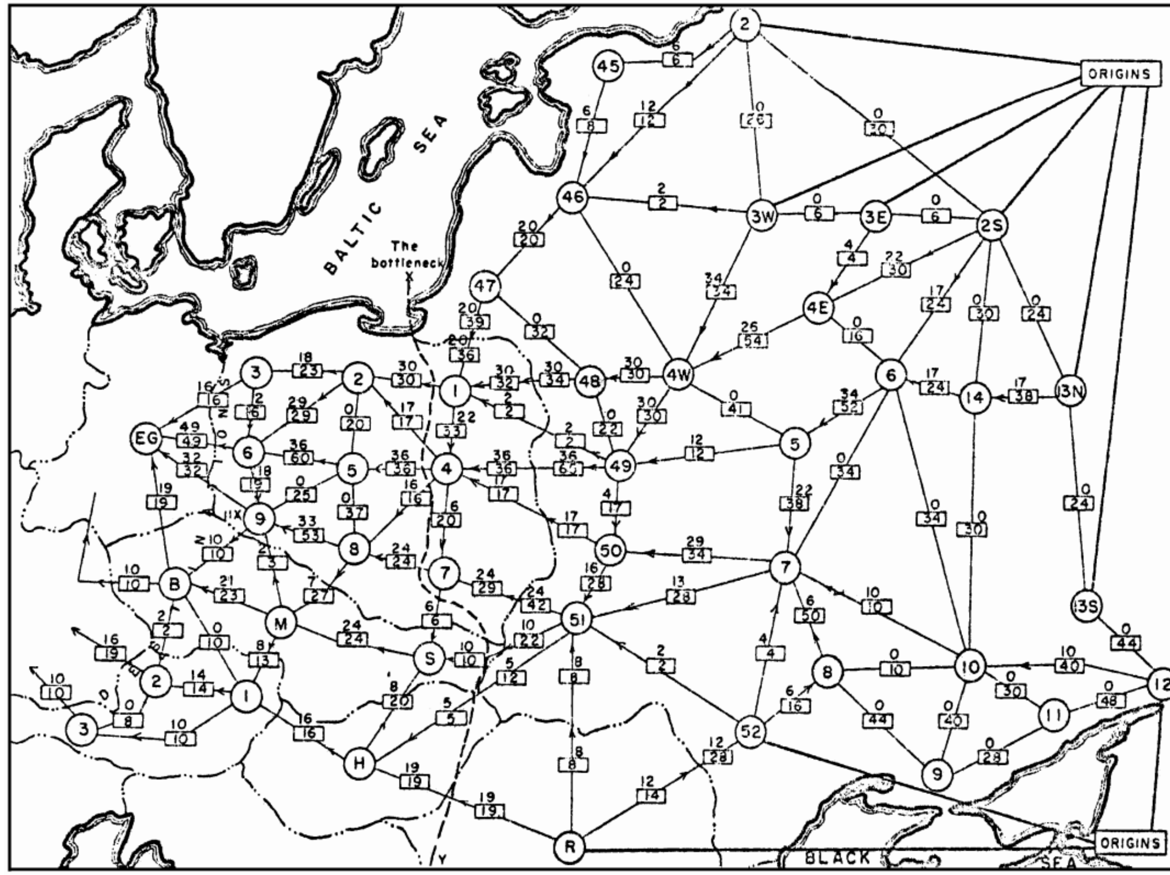
- The rabbit is the red “cut”!
- Any flow from S to T must cross the red curve.
- So it can have value at most 11.
- And here is the magic: such a proof is *always* possible!

Let’s try a more realistic graph



- You have 6 minutes! ;)

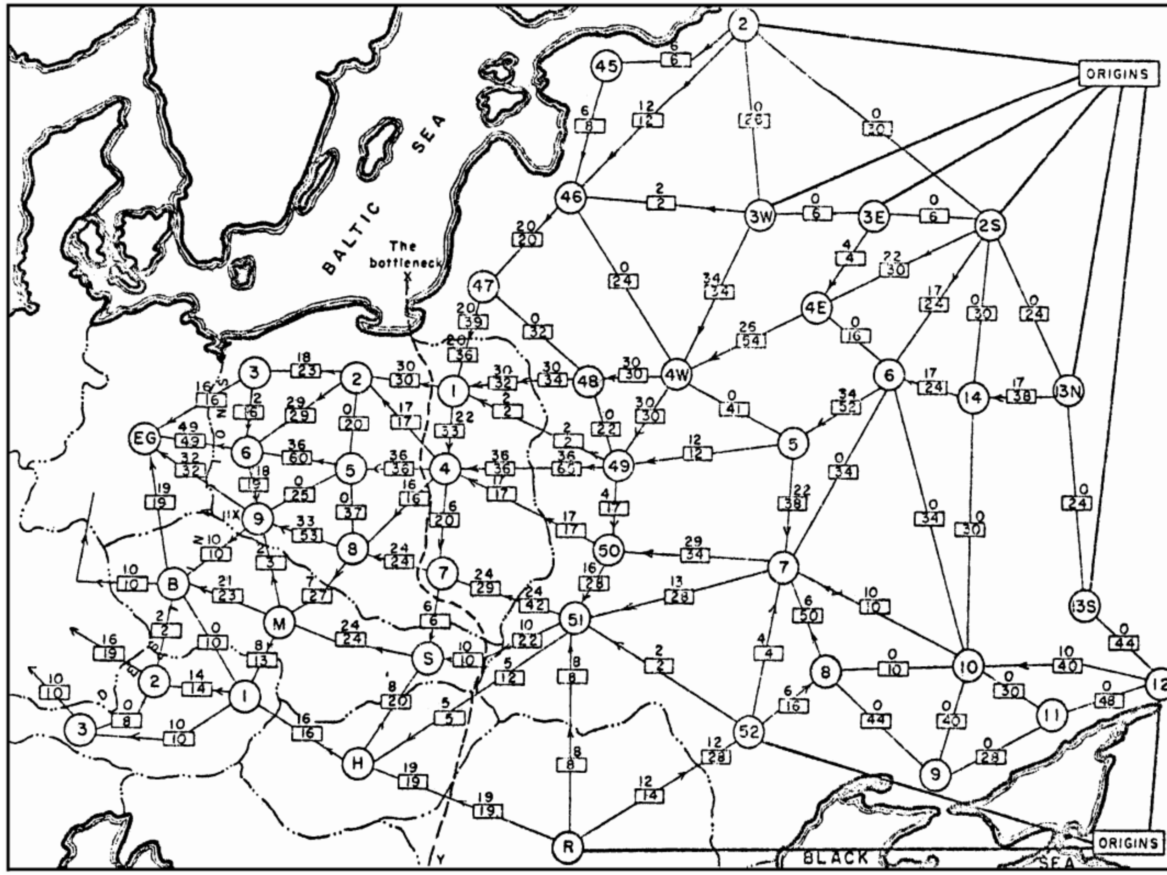
- How long do you think an optimization solver would take (on my laptop) to find the best solution here?
- How many lines of code do you think you have to write for it?
- How would someone who hasn't seen optimization approach this?



  - Trial and error?
  - Push a little flow here, a little there…
  - Do you think they are likely to find the best solution?

A bit of history behind this map

- From a secret report by Harris and Ross (1955) written for the Air Force.
- Railway network of the Western Soviet Union going to Eastern Europe.
- Declassified in 1999.
- Look at the min-cut on the map (called the “bottleneck”)!
- There are 44 vertices, 105 edges, and the max flow is 163K.



- Harris and Ross gave a heuristic which happened to solve the problem optimally in this case.
- Later that year (1955), the famous Ford-Fulkerson algorithm came out of the RAND corporation. The algorithm always finds the best solution (for rational edge costs).

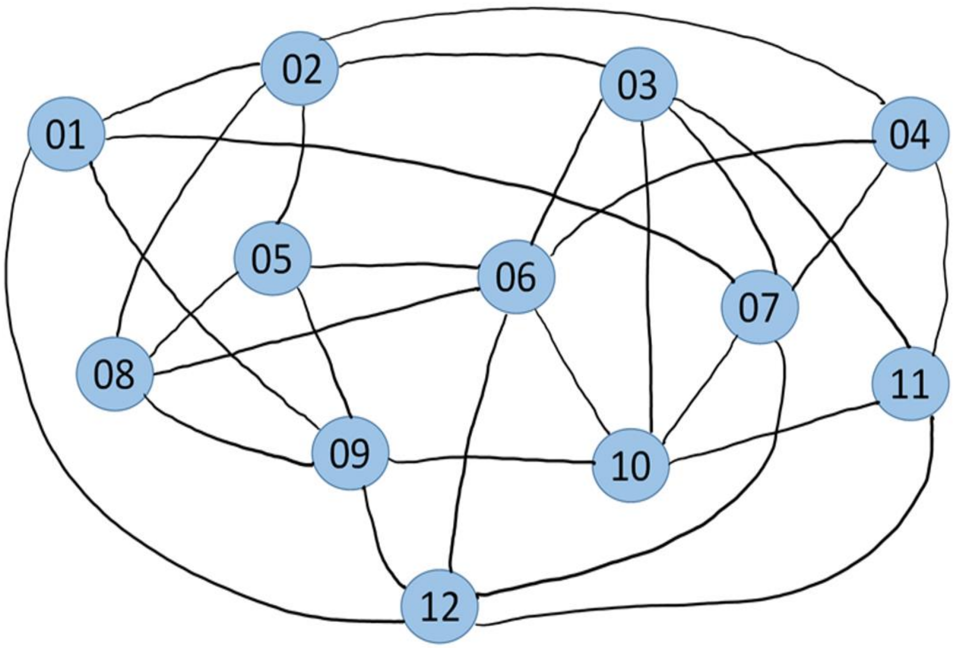
Let’s look at a second problem

…and tell me which one you thought was easier

Two finals in one day? No thanks.

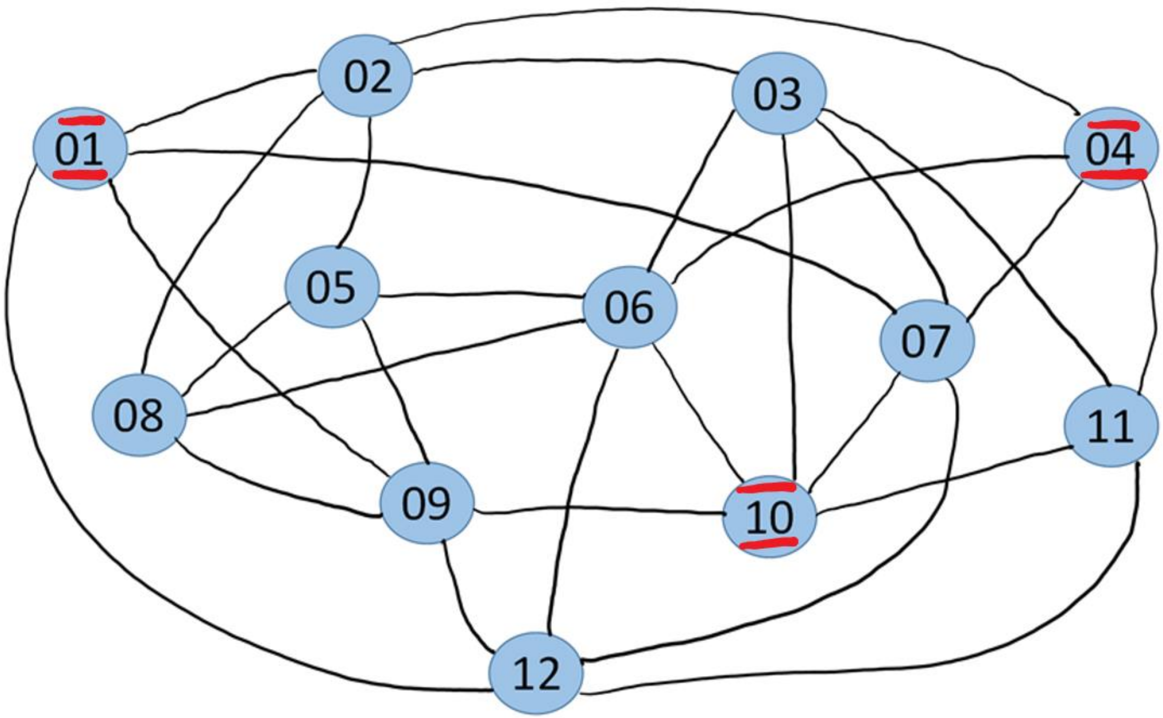
- The department chair at ORFE would like to schedule the final exams for 12 graduate courses offered this semester.
- He wants to have as many exams as possible on the same day, so everyone gets done quickly and goes on vacation.
- There is just one constraint:
No student should have >1 exam on that day.

- The nodes of this graph are the 12 courses.
- There is an edge between two nodes if and only if there is at least one student who is taking both courses.
- If we want to schedule as many exams as possible on the same day, what are we looking for in this graph?



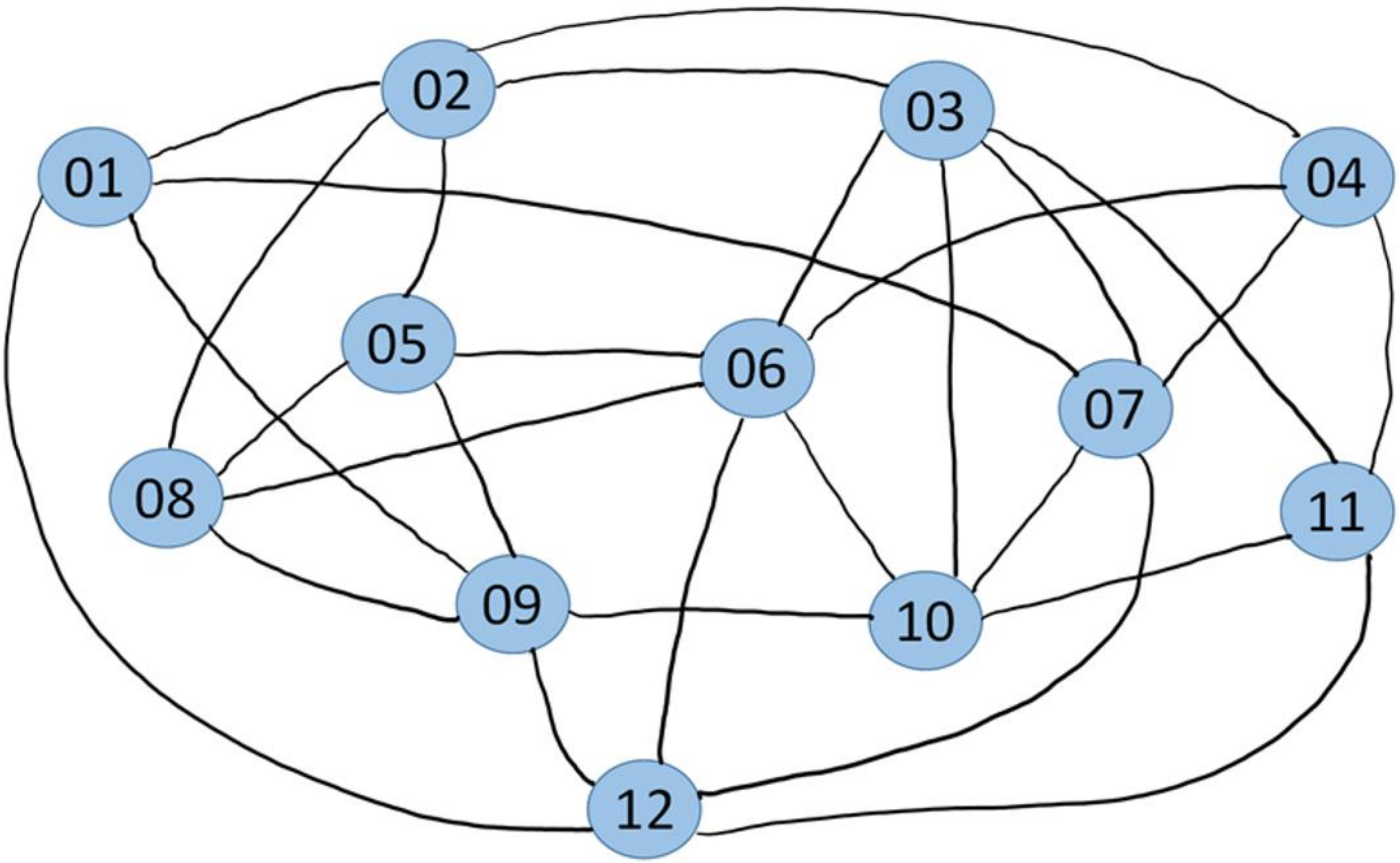
- The largest collection of nodes such that no two nodes share an edge.

- Let me start things off for you. Here is 3 concurrent final exams:

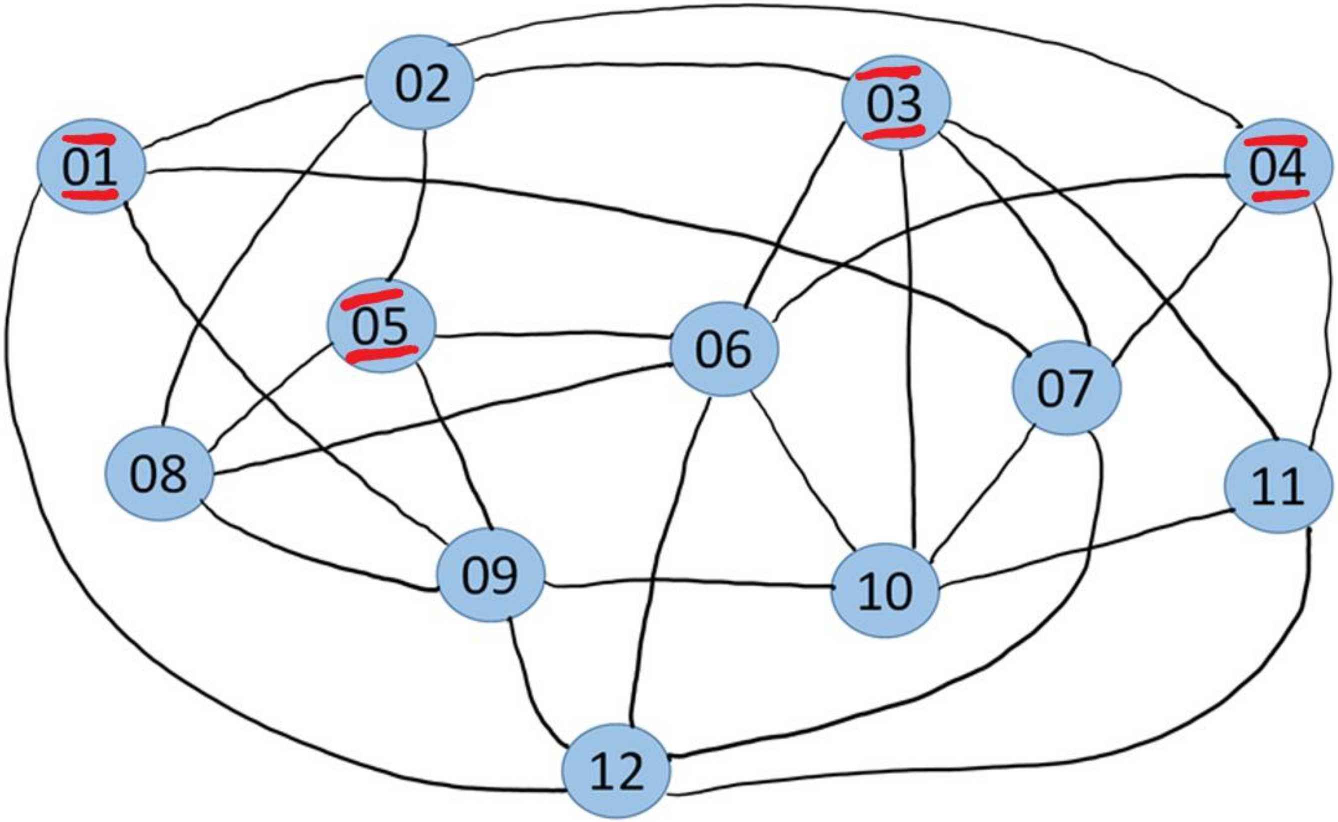


- Can you do better?
- How much better?
- You all get a copy of this graph.
  - You have 6 minutes!

You tell me, I draw…



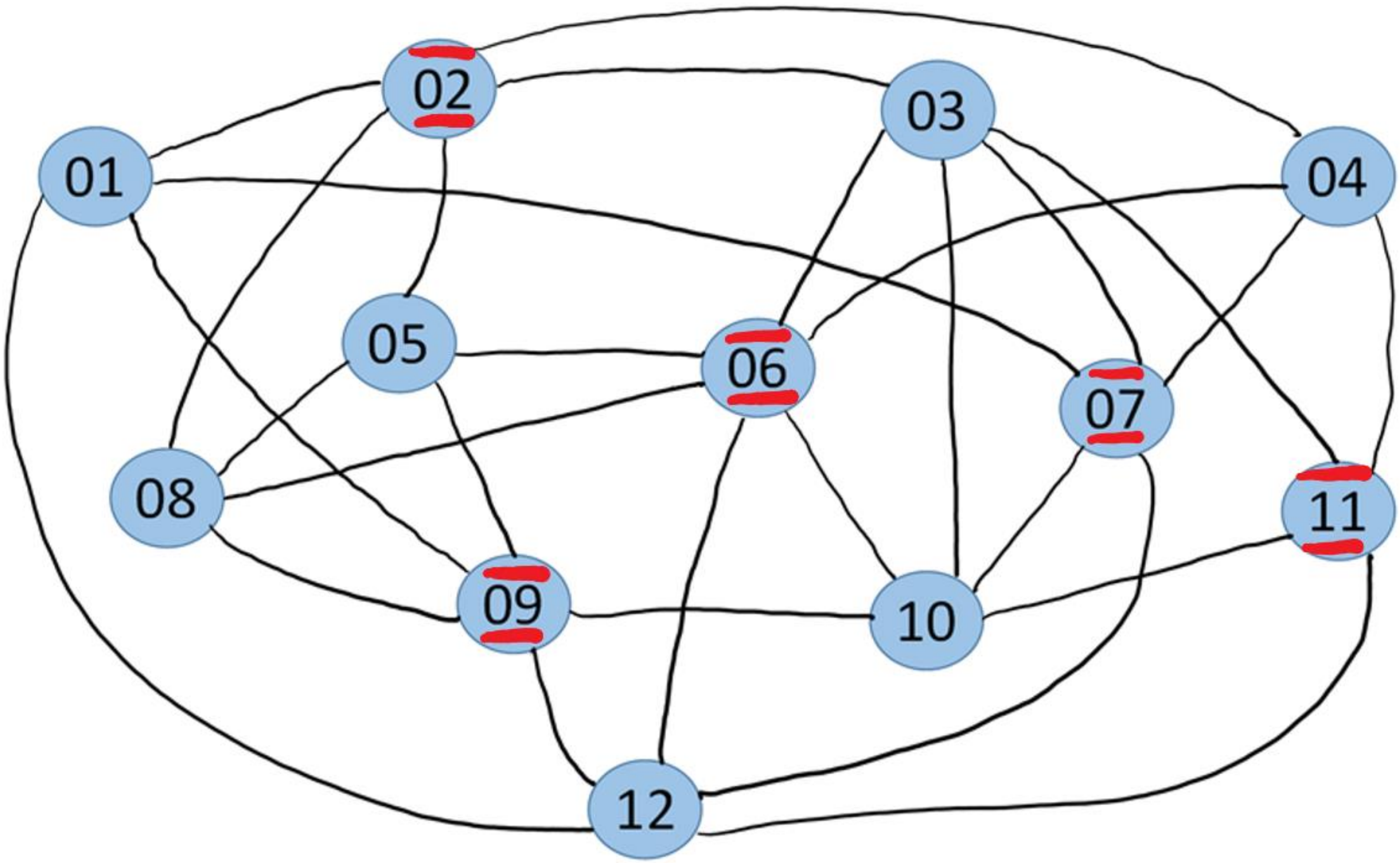
A couple of good attempts



4 exams

- Can you do better?

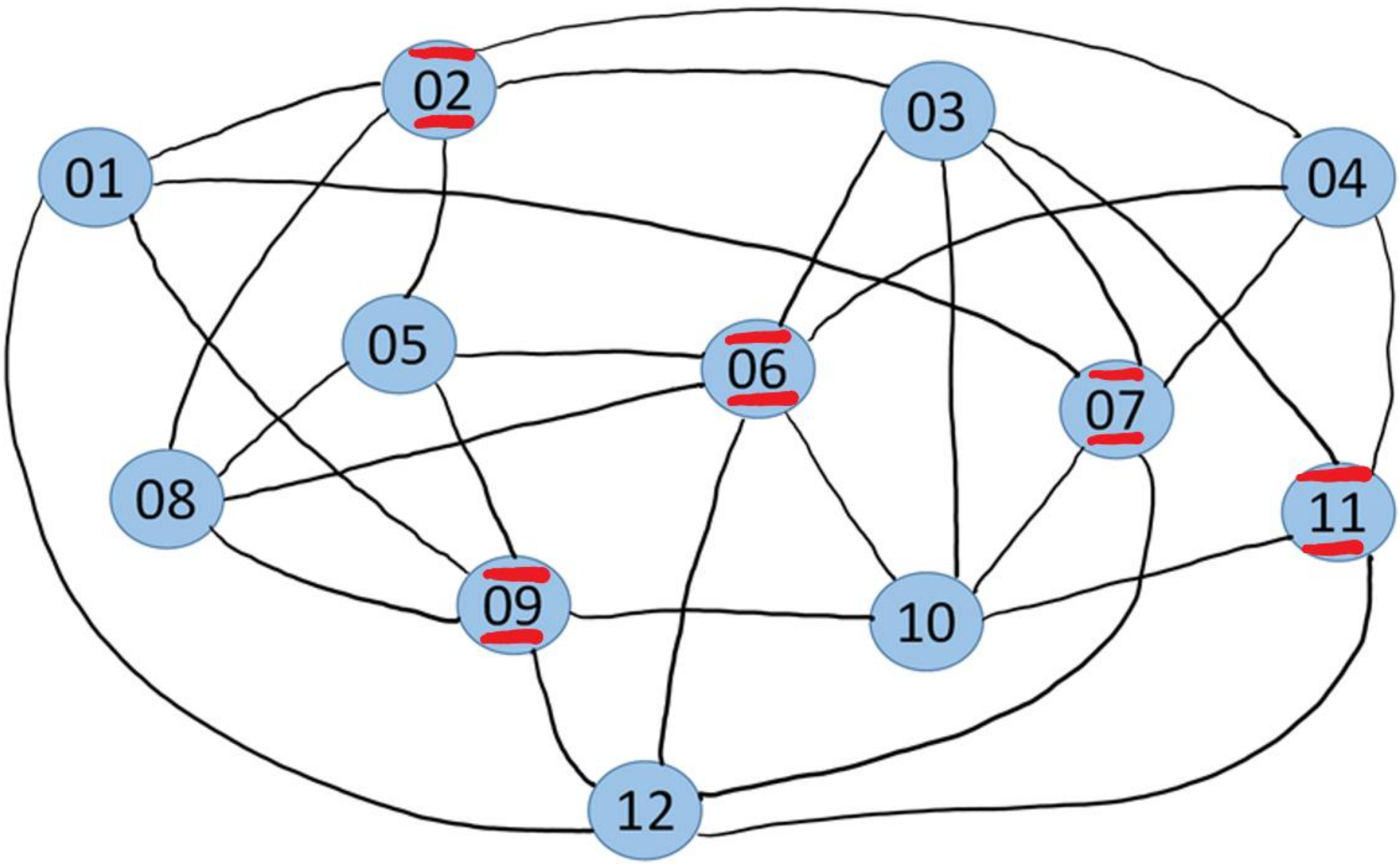
A couple of good attempts



5 exams

- Can you do better?

A couple of good attempts



5 exams

- Tired of trying?
- Is this the best possible?

5 is the best possible!

- Proof by magic?

- Unfortunately not ☹
- No magician in the world has pulled out such a rabbit to this day! (By this we mean a trick that would work on *all* graphs.)

- Of course there is always a proof:
  - Try all possible subsets of 6 nodes.
  - There are 924 of them.
  - Observe that none of them work.
- But this is no magic. It impresses nobody. We want a "short" proof. (We will formalize what this means.) Like the one in our max-flow example.
- Let's appreciate this further…

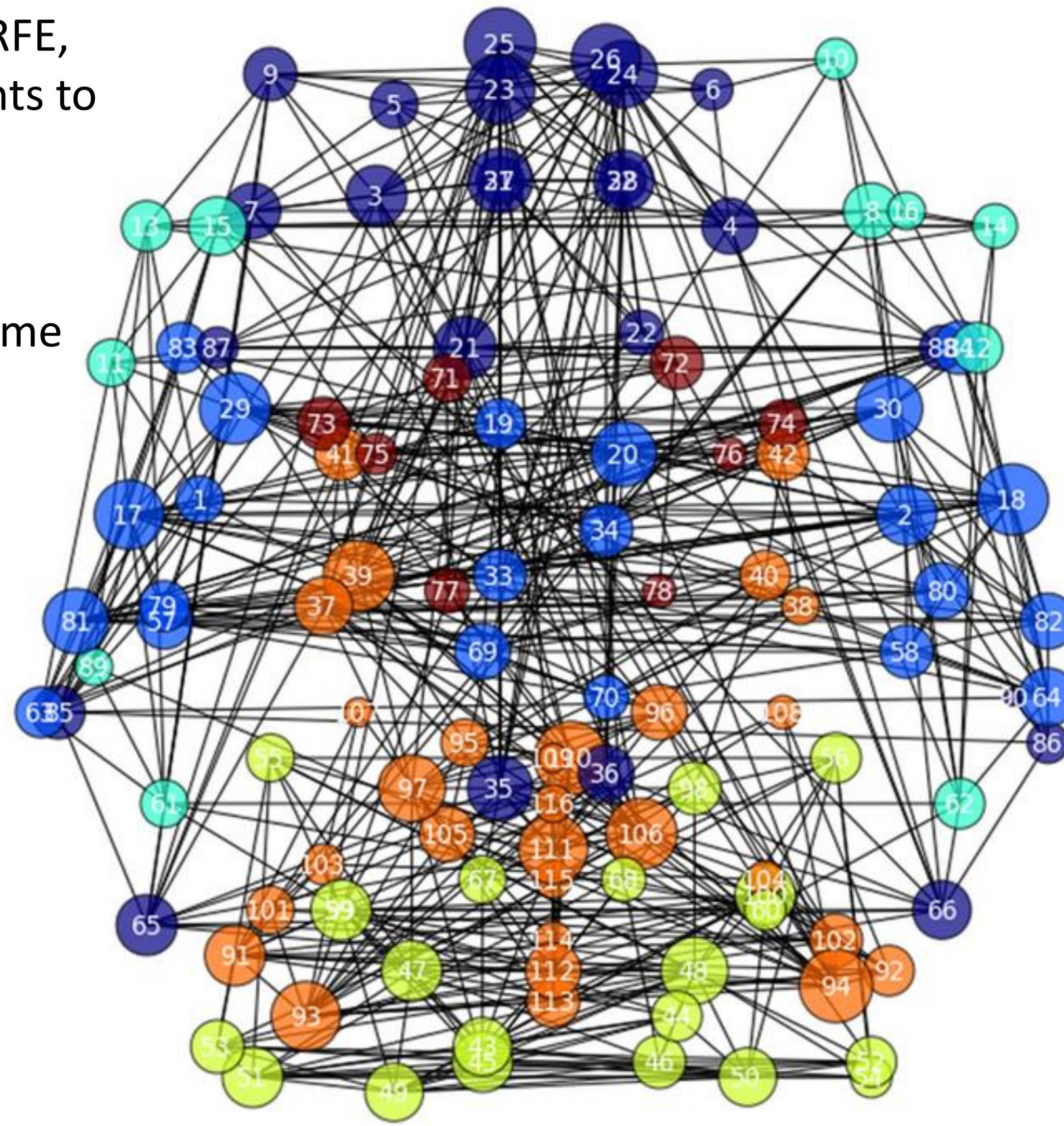
Let’s try another graph

- Encouraged by the success of ORFE, now the Dean of Engineering wants to the same for 115 SEAS courses.

- How many final exams on the same day are possible? Can you do 17?
    - You have 6 minutes! ;)

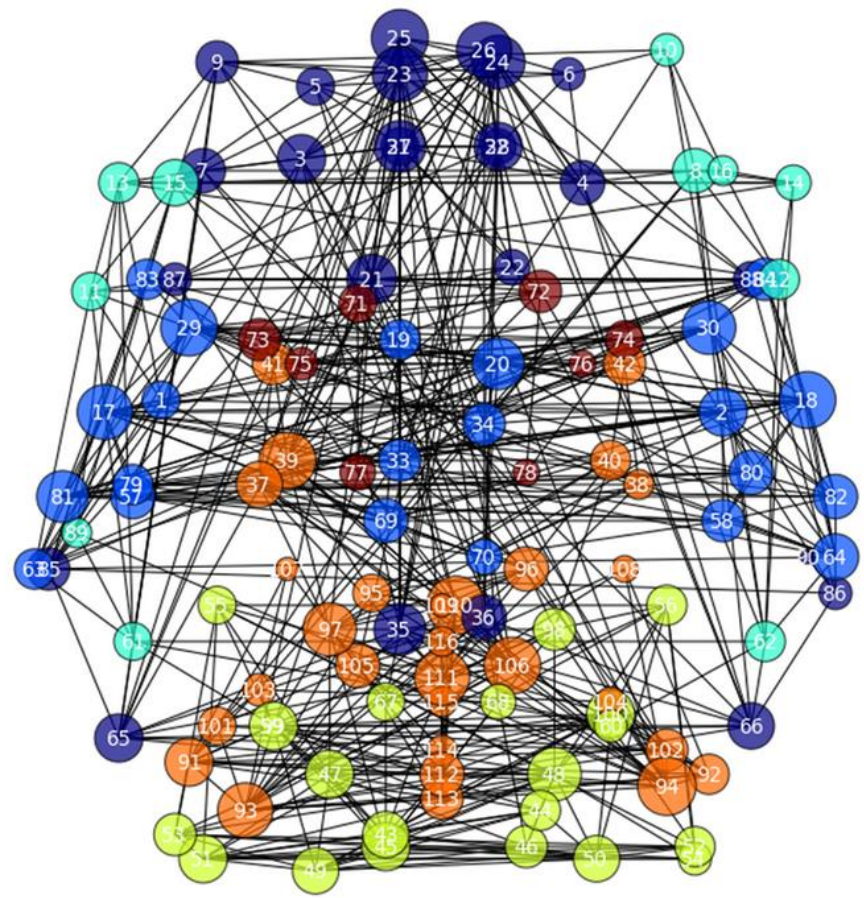
- Want to try out all possibilities for 17 exams?

- There are over 8000000000000000000 of them!



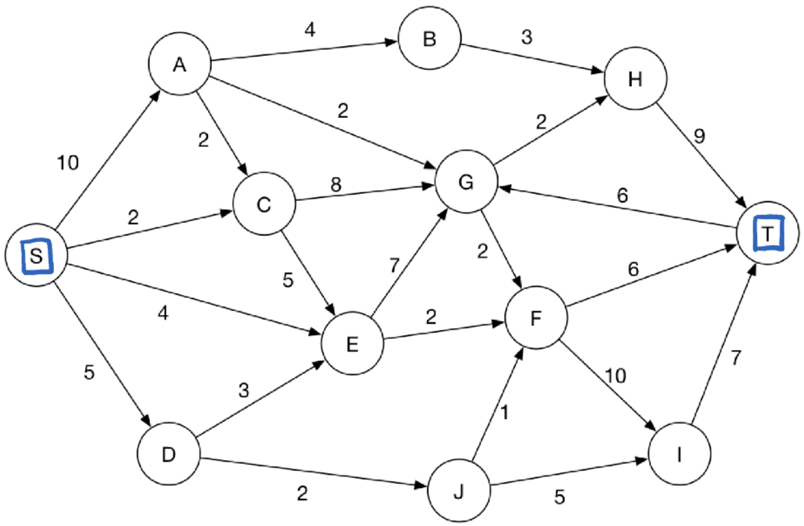
But there is some good news

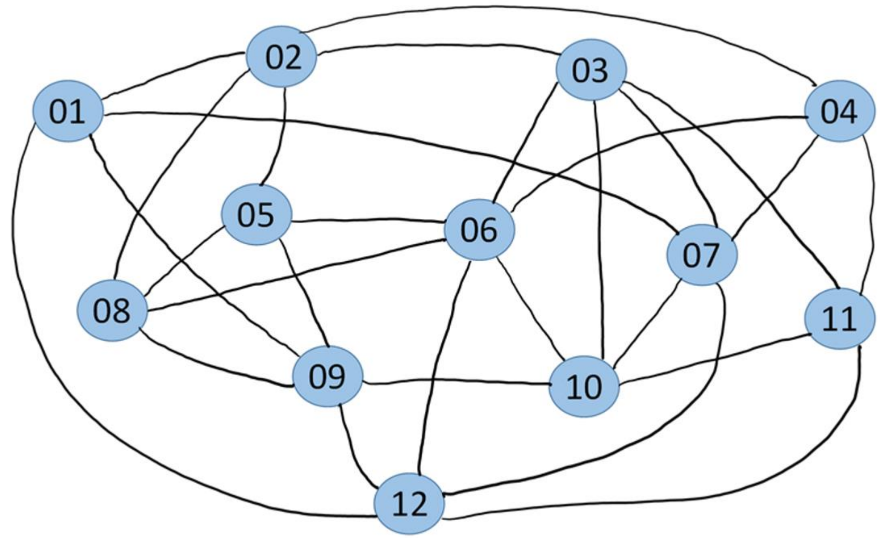
- Even though finding the best solution always may be too much to hope for, techniques from optimization (and in particular from the area of *convex optimization*) often allow us to find high-quality solutions with performance guarantees.
- For example, an optimization algorithm may quickly find 16 concurrent exams for you.
- You really want to know if 17 is impossible. Instead, another optimization algorithm (or sometimes the same one) tells you that 19 is impossible.



- This is very useful information! You know you got 16, and no one can do better than 19.
    - **We sill see a lot of convex optimization in this class!**

Which of the two problems was harder for you?

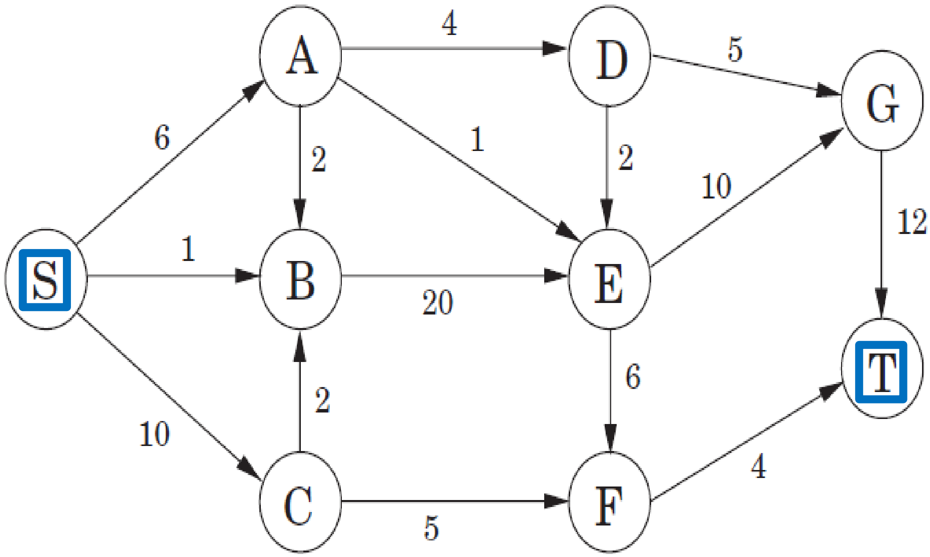




- Not always obvious. A lot of research in optimization and computer science goes into distinguishing the “tractable” problems from the “intractable” ones.
- The two brain teasers actually just gave you a taste of the P vs. NP problem. (If you have not heard about this, that’s OK. You will soon.)
- The first problem we can solve efficiently (in “polynomial time”).
- The second problem: no one knows. If you do, you literally get $1M!
  - More importantly, your algorithm immediately translates to an efficient algorithm for thousands of other problems no one knows how to solve.

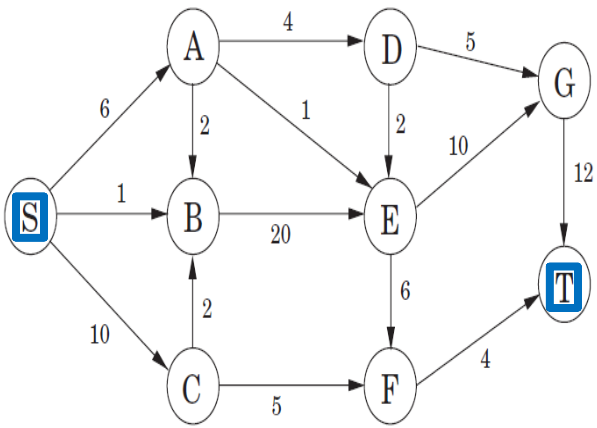
Modelling problems as a mathematical program

Let's revisit our first game



- What were your decision variables?
- What were your constraints?
- What was your objective function?

- **Rules of the game:**
  - Cannot exceed capacity on the edges.
  - For each node, except for S ant T, flow in = flow out (i.e., no storage).
  - **Goal:** ship as much oil as you can from S to T.



$$x_{SA},\ x_{AD},\ x_{BE},\ \dots,\ x_{GT}$$ ← Decision variables

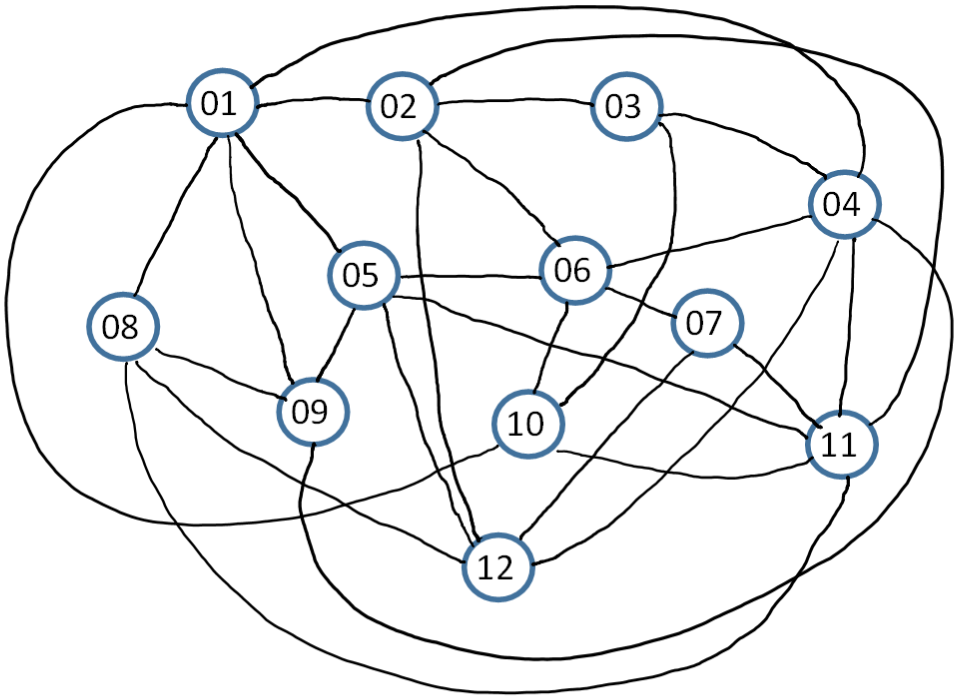
$$\max.\quad x_{SA} + x_{SB} + x_{SC}$$ ← Objective function

s.t.

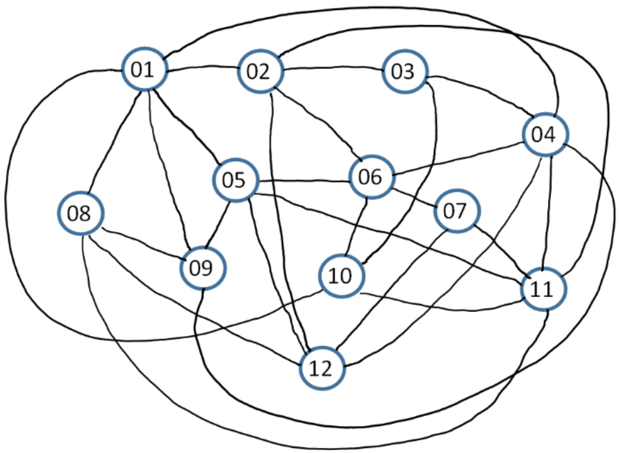
- $x_{SA},\ x_{AD},\ x_{BE},\ \dots,\ x_{GT} \geq 0$
- $x_{SA} \leq 6,\ x_{AB} \leq 2,\ x_{EG} \leq 10,\ \dots,\ x_{GT} \leq 12$
- $$\begin{cases} x_{SA} = x_{AD} + x_{AB} + x_{AE} \\ x_{SC} = x_{CB} + x_{CF} \\ \vdots \\ x_{CF} + x_{EF} = x_{FT}. \end{cases}$$

← Constraints

Let's revisit our second game



- What were your decision variables?
- What were your constraints?
- What was your objective function?



$x_1, x_2, \ldots, x_{12}$ ← Decision variables

$\max.\ x_1 + x_2 + \cdots + x_{12}$ ← Objective function

s.t.

- $x_i(1 - x_i) = 0, \quad i = 1, \ldots, 12$
- $\begin{cases} x_1 + x_2 \leq 1 \\ x_1 + x_8 \leq 1 \\ x_4 + x_6 \leq 1 \\ \vdots \\ x_{12} + x_8 \leq 1 \end{cases}$ (one per edge)

← Constraints

Why one hard and one easy? How can you tell?

$x_{SA}, x_{AD}, x_{BE}, \ldots, x_{GT}$

max. $x_{SA} + x_{SB} + x_{SC}$

s.t.

- $x_{SA}, x_{AD}, x_{BE}, \ldots, x_{GT} \geq 0$
- $x_{SA} \leq 6, x_{AB} \leq 2, x_{EG} \leq 10, \ldots, x_{GT} \leq 12$
- $\begin{cases} x_{SA} = x_{AD} + x_{AB} + x_{AE} \\ x_{SC} = x_{CB} + x_{CF} \\ \vdots \\ x_{CF} + x_{EF} = x_{FT}. \end{cases}$

$x_1, x_2, \ldots, x_{12}$

max. $x_1 + x_2 + \cdots + x_{12}$

s.t.

- $x_i(1 - x_i) = 0, \quad i = 1, \ldots, 12$
- $\begin{cases} x_1 + x_2 \leq 1 \\ x_1 + x_8 \leq 1 \\ x_4 + x_6 \leq 1 \\ \vdots \\ x_{12} + x_8 \leq 1 \end{cases}$ (one per edge)

- **Caution:** just because we can write something as a mathematical program, it doesn't mean we can solve it.

Fermat's Last Theorem

- Can you give me three positive integers x, y, z such that

$$x^2 + y^2 = z^2?$$

- Sure: (3, 4, 5) (5, 12, 13) (8, 15, 17) (7, 24, 25)
  (20, 21, 29) (12, 35, 37) (9, 40, 41) (28, 45, 53)

And there are infinitely many more…

- How about $x^3 + y^3 = z^3$?

- How about $x^4 + y^4 = z^4$?

- How about $x^5 + y^5 = z^5$?

Fermat's Last Theorem

- Fermat's conjecture (1637):

For $n \geq 3$, the equation $x^n + y^n = z^n$ has no solution over positive integers.

- Proved in 1994 (357 years later!) by Andrew Wiles.

(Was on the faculty in our math department until a few years ago.)


Arithmeticorum Liber II. 61

interuallum numerorum 2. minor autem 1 N. atque ideo maior 1 N. + 2. Oportet itaque 4 N. + 4. triplos esse ad 2. & adhuc superaddere 10. Ter igitur 2. adscitis vnitatibus 10. æquatur 4 N. + 4. & fit 1 N. 3. Erit ergo minor 3. maior 5. & satisfaciunt quæstioni.

IN QVAESTIONEM VII.

CONDITIONIS appositæ eadem rattio est quæ & appositæ præcedenti quæstioni, nil enim aliud requirit quàm vt quadratus interualli numerorum sit minor interuallo quadratorum, & Canones iidem hic etiam locum habebunt, vt manifestum est.

QVÆSTIO VIII.

PROPOSITVM quadratum diuidere in duos quadratos. Imperatum sit vt 16. diuidatur in duos quadratos. Ponatur primus 1 Q. Oportet igitur 16 − 1 Q. æquales esse quadrato. Fingo quadratum à numeris quotquot libuerit, cum defectu tot vnitatum quod continet latus ipsius 16. esto à 2 N. − 4. ipse igitur quadratus erit 4 Q. + 16. − 16 N. hæc æquabuntur vnitatibus 16 − 1 Q. Communis adiiciatur vtrimque defectus, & à similibus auferantur similia, fient 5 Q. æquales 16 N. & fit 1 N. 16/5 Erit igitur alter quadratorum 256/25. alter verò 144/25 & vtriusque summa est 400/25 seu 16. & vterque quadratus est.

OBSERVATIO DOMINI PETRI DE FERMAT.

CVbum autem in duos cubos, aut quadratoquadratum in duos quadratoquadratos & generaliter nullam in infinitum vltra quadratum potestatem in duos eiusdem nominis fas est diuidere cuius rei demonstrationem mirabilem sane detexi. Hanc marginis exiguitas non caperet.

QVÆSTIO IX.

RVRSVS oporteat quadratum 16 diuidere in duos quadratos. Ponatur rursus primi latus 1 N. alterius verò quotcunque numerorum cum defectu tot vnitatum, quot constat latus diuidendi. Esto itaque 2 N. − 4. erunt quadrati, hic quidem 1 Q. ille verò 4 Q. + 16. − 16 N. Cæterum volo vtrumque simul æquari vnitatibus 16. Igitur 5 Q. + 16. − 16 N. æquatur vnitatibus 16. & fit 1 N. 16/5 erit

H iii

Fermat's Last Theorem

- Fermat's conjecture (1637):

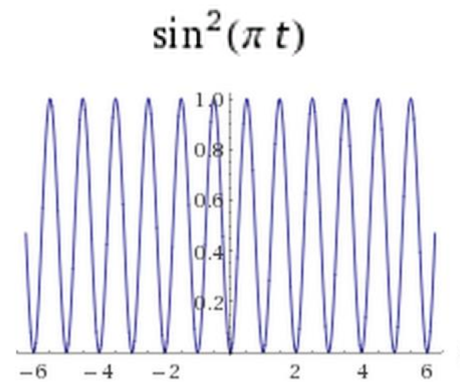
For $n \geq 3$, the equation $x^n + y^n = z^n$ has no solution over positive integers.

- Consider the following optimization problem (mathematical program):

$$\min_{x,y,z,n} \quad (x^n + y^n - z^n)^2$$

$$\text{s.t.} \quad x \geq 1,\ y \geq 1,\ z \geq 1,\ n \geq 3,$$

$$\sin^2 \pi n + \sin^2 \pi x + \sin^2 \pi y + \sin^2 \pi z = 0.$$



- Innocent-looking optimization problem: 4 variables, 5 constraints.

- If you could show the optimal value is non-zero, you would prove Fermat's conjecture!

Course objectives

- The skills I hope you acquire:

- Ability to view your own field through the lens of optimization and computation
    - To help you, we'll draw applications from operations research, statistics, finance, machine learning, engineering, …
- Learn about several topics in scientific computing
- More mathematical maturity and ability for rigorous reasoning
    - There will be some proofs in lecture. Easier ones on homework.
- Enhance your coding abilities (nothing too fancy, simple MATLAB)
    - There will be a MATLAB component on every homework and on the take-home final.
- Ability to recognize hard and easy optimization problems
- Ability to use optimization software
    - Understand the algorithms behind the software for some easier subclass of problems.

Software you need to download

- Right away:

MATLAB

Available for free to Princeton students

- In the next week or two (will appear on HW#2 or #3):

CVX

http://cvxr.com/cvx/

Course logistics

- See syllabus.
- Course website:

http://aaa.princeton.edu/orf363

- For those interested:
    - Princeton Optimization Seminar
    - http://orfe.princeton.edu/events/optimization-seminar

- Image credits and references:

- [DPV08] S. Dasgupta, C. Papadimitriou, and U. Vazirani. Algorithms. McGraw Hill, 2008.
- [Sch05] A. Schrijver. On the history of combinatorial optimization (till 1960). In "Handbook of Discrete Optimization", Elsevier, 2005. http://homepages.cwi.nl/~lex/files/histco.pdf