Lecture 1

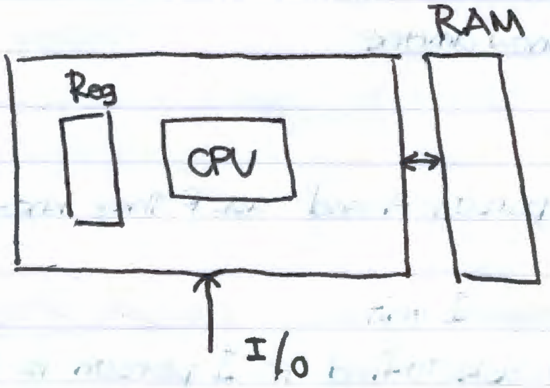
1/5/16

HW - 30%
MT - 30%
Final - 40%

Professor
Office Hours: Tues 7-8 AM
Thurs 11-12 AM
Boelter 3532C

Algorithm ↳ Al Khowrazmi Persian mathematician

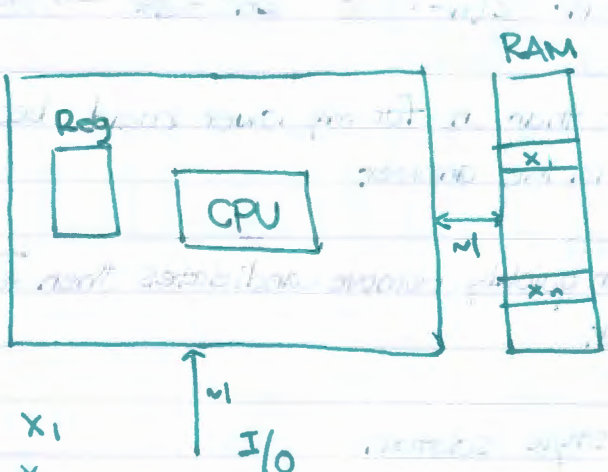
Universal Model of Computation (Serial/von Neumann)



- CPU - simple ops
- RAM - memory
- I/0 - input/output
- Limited # of local regs (16, 32, 64, etc.)

$\boxed{\text{ex:}}$ Given n integer numbers, how do we add them up? $\boxed{\text{Problem Statement}}$

Input sizes tend to be large, so we substitute n as a representation for any number



x_1
x_2
⋮
x_n

It takes about 1 unit of time.

unit's actual value depends on hardware available

Read ~2n
into CPU ~n
adding ~n-1
output ~1

Time complexity $\boxed{\sim 4n}$

Lower bound analysis: can we do any better?

Informally we see that to sum n items we must read n items which is order n, hence our algorithm with $O(4n)$ is good!

formal definition

ex: Famous: everyone knows

he/she doesn't know anyone

Model of computation:

Pick 2 people: ask person A and ask if they know person B.

$$A \xrightarrow{\text{know}} B \qquad 1 \text{ unit}$$

It would take $\sim 2(n-1)$ asks to find if 1 person is famous. Note that if someone is famous, no one else can be famous.

$\sim n-1$ others all know F

$\sim n-1$ F does not know any others

So the time complexity is $n \cdot 2(n-1) = 2n^2 - 2n \rightarrow \sim 2n^2 \rightarrow \sim n^2$

Informally, I can't do better than n for my lower bound, because each person can have a meaningful part in the answer.

Problem Reduction: If I can quickly remove candidates then I have the potential to improve the answer.

Let's try a tournament style solution

Random – truly want a random person

Arbitrary – doesn't matter which person

Key property

$A \xrightarrow{\text{know}} B$ < yes / no

With 1 question, I can eliminate 1 famous candidate.

After $n-1$ questions I have only 1 candidate.
then it's simply $2(n-1)$ for this candidate, and we will know either that this remaining person is famous or there is no famous person.

$\approx 3(n-1)$

$\sim n$

Asymptotic Analysis - formal analysis of effectiveness

		10
	n	10
polynomial	$2n$	20
	n^2	100
exponential	2^n	1024
	$n!$	4m

$f(n) = O(g(n))$

if $\exists$ ~~non zero c s.t.~~ n_0, c such that

$f(n) \leq c\, g(n)$ for $n \geq n_0$

"The same or less, considering constants"

ex:

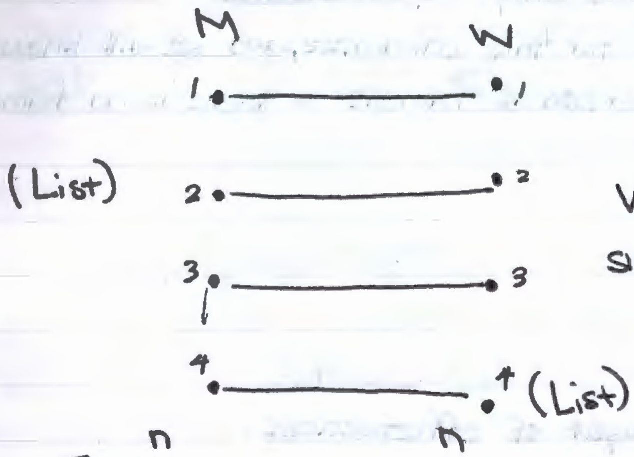
$f(n) = O(n^2)$

$\frac{n^2}{4}$ ✓ — $5n^2$ ✓ make c compatible — $\log n$ ✓ it is less, so it still works — 2^n ✗ nope.jpg there is no c that works

Lecture 2: 1/7/16

Matching Problems

- Primarily 1 to 1



We want a perfect match, that is, every person needs a match.

With the absence of other criteria, we can simply draw horizontal lines.

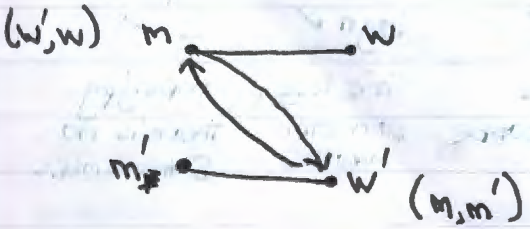
Fig 1.

However, we additionally have an ordering list of preferences for each person (every person is ranked, no ties).

Even given this set of lists, without additional requirements the above solution still works.

Convention: matches are horizontal parts of the graph

If we have



This matching is called unstable

New Problem:

- Given a situation like in Fig 1, I want
  - A perfect match
  - A stable match.

The Algorithm

while (not all men matched) Begin with no matches
- Pick any man
  - He goes to his highest priority woman and proposes
    - If woman is single, she will accept
    - If man is higher on list than the woman's current match, she will dump her man and accept the new man.

Let's Observe some properties of the algorithm....
- A man's match will worsen as he gets dumped
- A woman's match will improve as she gets ~~to~~ proposed to.
- Furthermore, once a woman is matched, she will remain matched for the algorithm.

Claim - At the end of the algorithm, we have a stable matching

Proof by contradiction - There is an unstable matching at the end of the algorithm.

Two cases:

1) m did not propose to w'
2) m did propose to w'

1) If m did not propose to w' and ended up with w, then w cannot be of lower preference than w', so there is a contradiction.
2) If m did propose to w' and w' was with m'', then since m wasn't accepted then m'' > m in w''s opinion. HOWEVER, m > m' and w' ended up with m' so we have a contradiction.

~~m' > w~~ m' > m and m > m' contradiction