15-252
More Great Ideas in Theoretical Computer Science

Lecture 1:
Sorting Pancakes

January 19th, 2018

Question

If there are **n** pancakes in total (all in different sizes), what is the max number of "flips" that we would ever have to use to sort them?

This description is a bit ambiguous.

$P_n =$ the number described above

What is P_n ?

Understanding the question

$$P_n = \max_S \min_A \ \# \text{ flips when sorting } S \text{ by } A$$

over all strategies/algorithms for sorting

over all pancake stacks of size **n**

Number of flips necessary to sort the worst stack of size **n**.

Is it always possible to sort the pancakes?

Yes!

A sorting strategy (algorithm):

- Move the largest pancake to the bottom.
- Recurse on the other n-1 pancakes.

Playing around with an example

Introducing notation:

- represent a pancake with a number from 1 to n.
- represent a stack as a permutation of {1,2,…,n}
  e.g. (5 2 3 4 1)

top bottom

Let X = min number of flips to sort (5 2 3 4 1)

What is X ?

Playing around with (5 2 3 4 1)

$$? \leq X \leq 4$$

Need an argument for a lower bound.

A strategy/algorithm for sorting gives us an upper bound.

$0 \leq X$?

$1 \leq X$?

$2 \leq X$?

$3 \leq X$?

$4 \leq X$?

Playing around with (5 2 3 4 1)

Proposition: $X = 4$

Proof: We already showed $X \leq 4$.

We now show $X \geq 4$. The proof is by contradiction.

So suppose we can sort the pancakes using 3 or less flips.

Observation: Right before a pancake is placed at the bottom of the stack, it must be at the top.

Claim: The first flip must put 5 on the bottom of the stack.
Proof: If the first flip does not put 5 on the bottom of the stack, then it puts it somewhere in the middle of the stack.
After 3 flips, 5 must be placed at the bottom.
Using the observation above, 2nd flip must send 5 to the top.
Then after 2 flips, we end up with the original stack.
But there is no way to sort the original stack in 1 flip.
The claim follows. □

Playing around with (5 2 3 4 1)

Proposition: $X = 4$

Proof continued:

So we know the first flip must be: $(5\ 2\ 3\ 4\ 1) \longrightarrow (1\ 4\ 3\ 2\ 5)$.

In the remaining 2 flips, we must put 4 next to 5.

Obviously 5 cannot be touched.

So we can ignore 5 and just consider the stack $(1\ 4\ 3\ 2)$.

We need to put 4 at the bottom of this stack in 2 flips.

Again, using the observation stated above,
the next two moves must be:

$$(1\ 4\ 3\ 2) \longrightarrow (4\ 1\ 3\ 2) \longrightarrow (2\ 3\ 1\ 4)$$

This does not lead to a sorted stack,
which is a contradiction since we assumed we could sort the stack
in 3 flips. $\square$

Playing around with (5 2 3 4 1)

$$X = 4$$

What does this say about P_n ?

Pick the one that you think is true:

$P_n = 4$

$P_n \leq 4$

$P_n \geq 4$

$P_5 = 4$

$P_5 \leq 4$

$P_5 \geq 4$

None of the above.

Beats me.

Playing around with (5 2 3 4 1)

$$X = 4$$

What does this say about P_n ?

$$P_5 = \max_S \min_A \ \# \text{ flips when sorting } S \text{ by } A$$

↳ all stacks of size 5

all stacks:	$(5\ 2\ 3\ 4\ 1)$	$(5\ 4\ 3\ 2\ 1)$	$(1\ 2\ 3\ 4\ 5)$	$(5\ 4\ 1\ 2\ 3)$	$\cdots$
min # flips:	4	1	0	2	

$P_5 =$ max among these numbers

So: $X = 4 \implies P_5 \geq 4$

Playing around with (5 2 3 4 1)

In fact: (will not prove) $P_5 = 5$

$$5 \leq P_5 \leq 5$$

Find a specific "hard" stack.
Show any method
must use 5 flips.

Find a generic method
that sorts any 5-stack
with 5 flips.

Good progress so far:

- we understand the problem better
- we made some interesting observations

Ok what about P_n for general n ?

P_n for small n

$$P_0 = 0$$

$$P_1 = 0$$

$$P_2 = 1$$

$$P_3 = 3$$

upper bound:

- bring largest to the bottom in 2 flips
- sort the other 2 in 1 flip (if needed)

lower bound:

(1 3 2) requires 3 flips.

A general upper bound: “Bring-to-top” alg.

```
if n = 1: do nothing
else:
    - bring the largest pancake to bottom in 2 flips
    - recurse on the remaining n-1 pancakes
```

A general upper bound: “Bring-to-top” alg.

```
if n = 1:  do nothing
else if n = 2:  sort using at most 1 flip
else:
    - bring the largest pancake to bottom in 2 flips
    - recurse on the remaining n-1 pancakes
```

$T(n)$ = max # flips for this algorithm

$$T(1) = 0$$

$$T(2) \leq 1$$

$$T(n) \leq \boxed{2} + \boxed{T(n-1)} \qquad \text{for } n \geq 3$$

$$\Longrightarrow \quad T(n) \leq 2n - 3 \qquad \text{for } n \geq 2$$

A general upper bound: “Bring-to-top” alg.

Theorem: $P_n \leq 2n - 3 \quad \text{for } n \geq 2\,.$

Corollary: $P_3 \leq 3\,.$

Corollary: $P_5 \leq 7\,.$

(So this is a loose upper bound, i.e. not tight.)

A general lower bound

How about a lower bound?

What is the worst initial stack?

You must argue against **all** possible strategies.

A general lower bound

Observation:

Given an initial stack, suppose pancakes i and j are adjacent.

They will remain adjacent if we never insert the spatula in between them.

$$(\boxed{5\ 2}\ 3\ 4\ 1)$$

So:

If i and j are adjacent and $|i - j| > 1$,
then we **<u>must</u>** insert the spatula in between them.

Definition:

We call i and j a **bad** pair if

- they are adjacent
- $|i - j| > 1$

A general lower bound

Lemma (Breaking-apart argument):

A stack with b bad pairs needs at least b flips to be sorted.

e.g. $(\boxed{5\ 2}\,3\,\boxed{4\ 1})$ requires at least 2 flips.

In fact, we can conclude it requires at least 3 flips. Why?

Bottom pancake and plate can also form a bad pair.

A general lower bound

Theorem: $P_n \geq n$ for $n \geq 4$.

Proof:

Take cases on the parity of n.

If n is even, the following stack has n bad pairs:

$$(2\ 4\ 6\ \cdots\ n-2\ n\ 1\ 3\ 5\ \cdots\ n-1)$$

If n is odd, the following stack has n bad pairs:

$$(1\ 3\ 5\ \cdots\ n-2\ n\ 2\ 4\ 6\ \cdots\ n-1)$$

By the previous lemma, both need n flips to be sorted.

So $P_n \geq n$ for $n \geq 4$. □

Where did we use the assumption $n \geq 4$?

So what were we able to prove about P_n ?

Theorem: $n \leq P_n \leq 2n - 3$ for $n \geq 4.$

Best known bounds for P_n

Jacob Goodman 1975: what we saw

published under pseudonym Harry Dweighter

William Gates and Christos Papadimitriou 1979:

$$\frac{17}{16}n \leq P_n \leq \frac{5}{3}(n+1)$$

Currently best known: $$\frac{15}{14}n \leq P_n \leq \frac{18}{11}n$$

BP_n

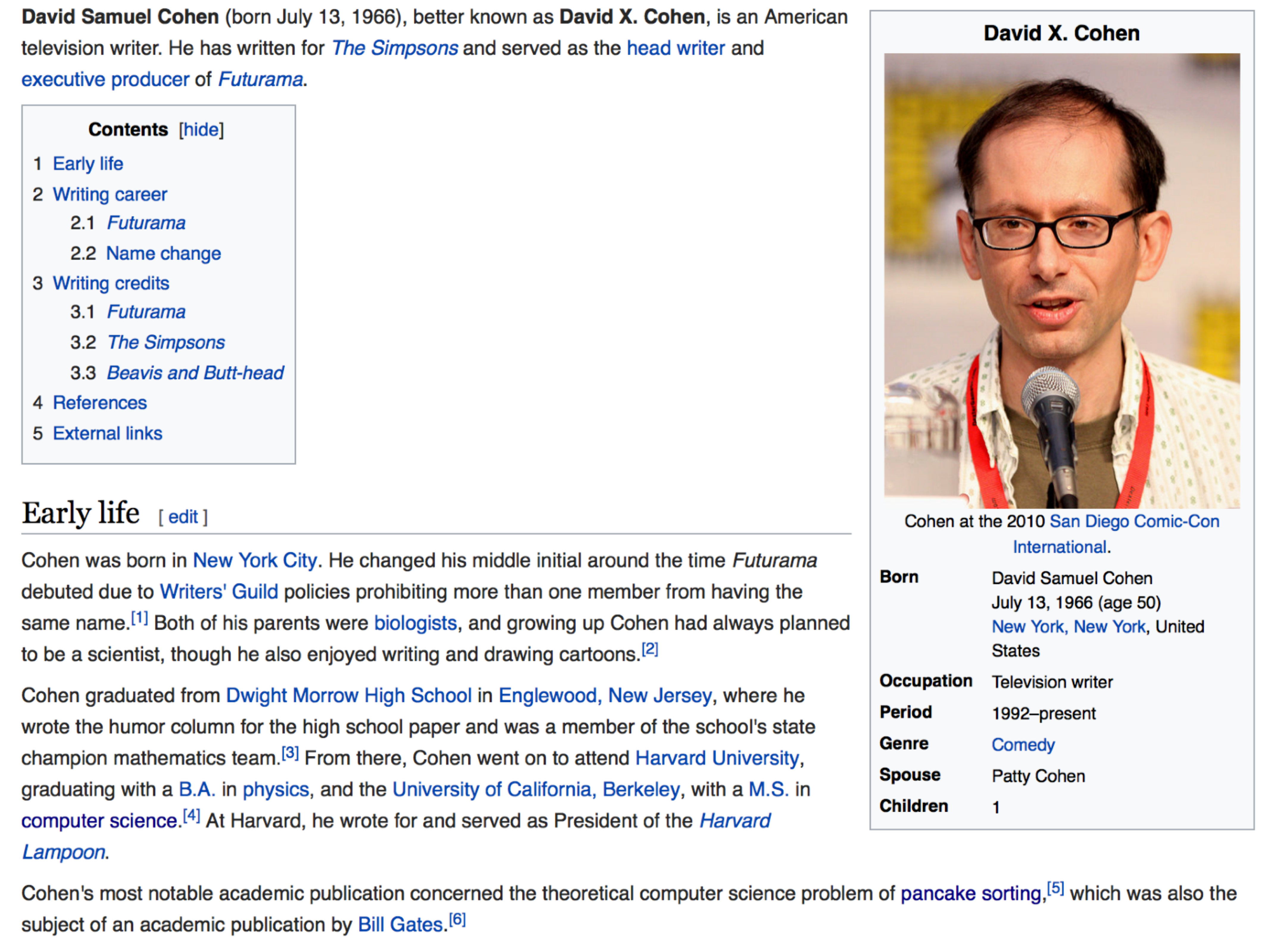
William Gates and Christos Papadimitriou 1979:

Introduced “Burnt pancakes” problem.

$$\frac{3}{2}n - 1 \leq BP_n \leq 2n + 3$$

David Cohen and Manuel Blum 1995:

$$\frac{3}{2}n \leq BP_n \leq 2n - 2$$



David Samuel Cohen (born July 13, 1966), better known as **David X. Cohen**, is an American television writer. He has written for *The Simpsons* and served as the head writer and executive producer of *Futurama*.

David X. Cohen

Cohen at the 2010 San Diego Comic-Con International.

Born	David Samuel Cohen July 13, 1966 (age 50) New York, New York, United States
Occupation	Television writer
Period	1992–present
Genre	Comedy
Spouse	Patty Cohen
Children	1

Early life [edit]

Cohen was born in New York City. He changed his middle initial around the time *Futurama* debuted due to Writers' Guild policies prohibiting more than one member from having the same name.[1] Both of his parents were biologists, and growing up Cohen had always planned to be a scientist, though he also enjoyed writing and drawing cartoons.[2]

Cohen graduated from Dwight Morrow High School in Englewood, New Jersey, where he wrote the humor column for the high school paper and was a member of the school's state champion mathematics team.[3] From there, Cohen went on to attend Harvard University, graduating with a B.A. in physics, and the University of California, Berkeley, with a M.S. in computer science.[4] At Harvard, he wrote for and served as President of the *Harvard Lampoon*.

Cohen's most notable academic publication concerned the theoretical computer science problem of pancake sorting,[5] which was also the subject of an academic publication by Bill Gates.[6]

Manuel Blum (Caracas, 26 April 1938) is a Venezuelan computer scientist who received the Turing Award in 1995 "In recognition of his contributions to the foundations of computational complexity theory and its application to cryptography and program checking".[2][3][4][5][6][7][8]

Education [edit]

Blum was educated at MIT, where he received his bachelor's degree and his master's degree in EECS in 1959 and 1961 respectively, and his Ph.D. in mathematics in 1964 supervised by Marvin Minsky.[1][7]

Career [edit]

He worked as a professor of computer science at the University of California, Berkeley until 1999. In 2002 he was elected to the United States National Academy of Sciences.

He is currently the Bruce Nelson Professor of Computer Science at Carnegie Mellon University, where his wife, Lenore Blum,[9] and son, Avrim Blum, are also professors of Computer Science.

Research [edit]

In the 60s he developed an axiomatic complexity theory which was independent of concrete machine models. The theory is based on Gödel numberings and the Blum axioms. Even though the theory is not based on any machine model it yields concrete results like the compression theorem, the gap theorem, the honesty theorem and the Blum speedup theorem.

Manuel Blum

Manuel Blum (left) with his wife Lenore Blum and their son Avrim Blum, 1973

Born	April 26, 1938 (age 79) Caracas, Venezuela
Residence	Pittsburgh
Fields	Computer Science
Institutions	University of California, Berkeley Carnegie Mellon University
Alma mater	Massachusetts Institute of Technology
Thesis	*A Machine-Independent Theory of the Complexity of Recursive Functions* (1964)
Doctoral advisor	Marvin Minsky[1]
Doctoral students	Leonard Adleman Dana Angluin C. Eric Bach Joan Boyar William Evans Peter Gemmell John Gill, III Shafi Goldwasser

Best known bounds for P_n

n	P_n
4	4
5	5
6	7
7	8
8	9
9	10
10	11
11	13
12	14
13	15
14	16
15	17
16	18
17	19
18	20
19	22

$P_{20} = ?$

23 or 24

Why study pancake numbers?

Perhaps surprisingly, it has interesting applications.

- In designing efficient networks that are resilient to failures of links.

  Google "pancake network"

- In biology.

  Can think of chromosomes as permutations.

  Interested in mutations in which some portion of the chromosome gets flipped.

Lessons

Simple problems may be hard to solve.

Simple problems may have far-reaching applications.

By studying pancakes, you can be a billionaire.

Analogy with computation

input: initial stack

output: sorted stack

computational problem: (input, output) pairs
pancake sorting problem

computational model: specified by the allowed operations on the input.

algorithm: a precise description of how to obtain the output from the input.

computability: is it always possible to sort the stack?

complexity: how many flips are needed?