An experimental investigation on the pancake problem

Bruno Bouzy

Paris Descartes University

IJCAI Computer Game Workshop

Buenos-Aires – July 26, 2015

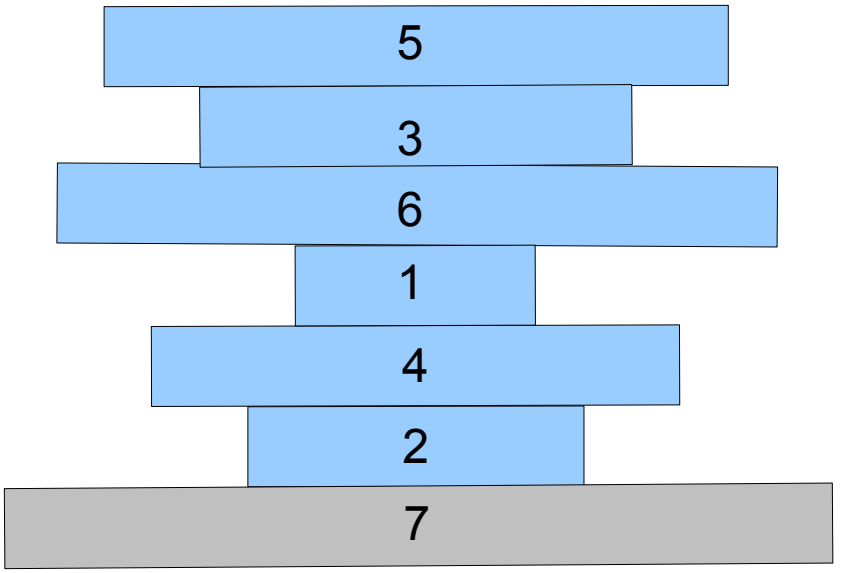
Outline

- The pancake problem
- The genome rearrangement problem
- MCS and IDDFS
- R-approx
- EffSort, AlternateSort, BREF, FG.
- Experimental results
- Conclusions

Example

A stack of pancakes.

Each pancake has a size.

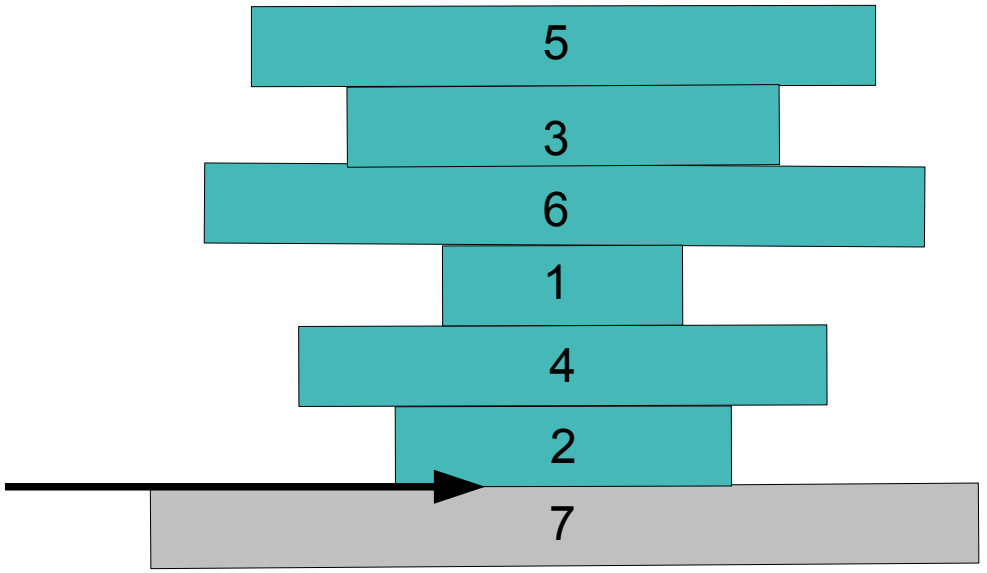


Goal : sort the stack with the greatest pancake at the bottom.

Example

Action :

insert a spatula below a pancake and flip the sub-stack above the spatula.

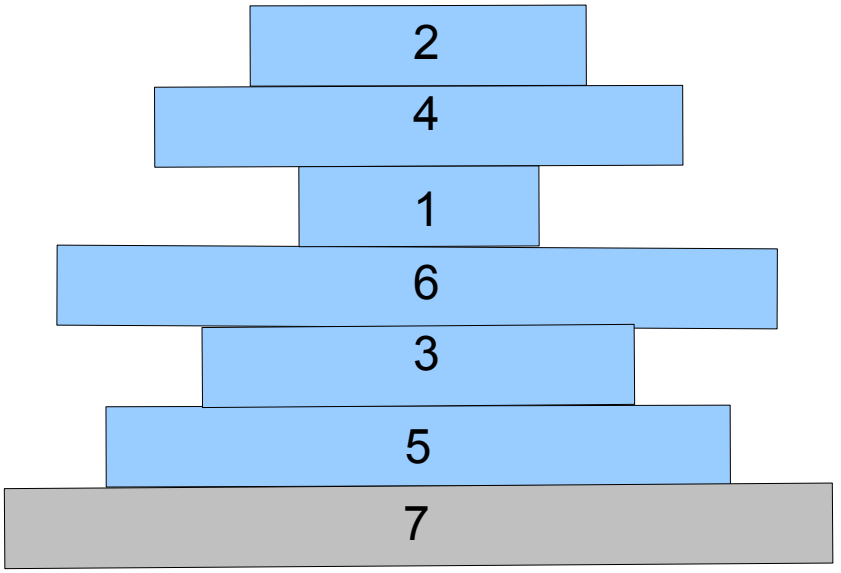


Example

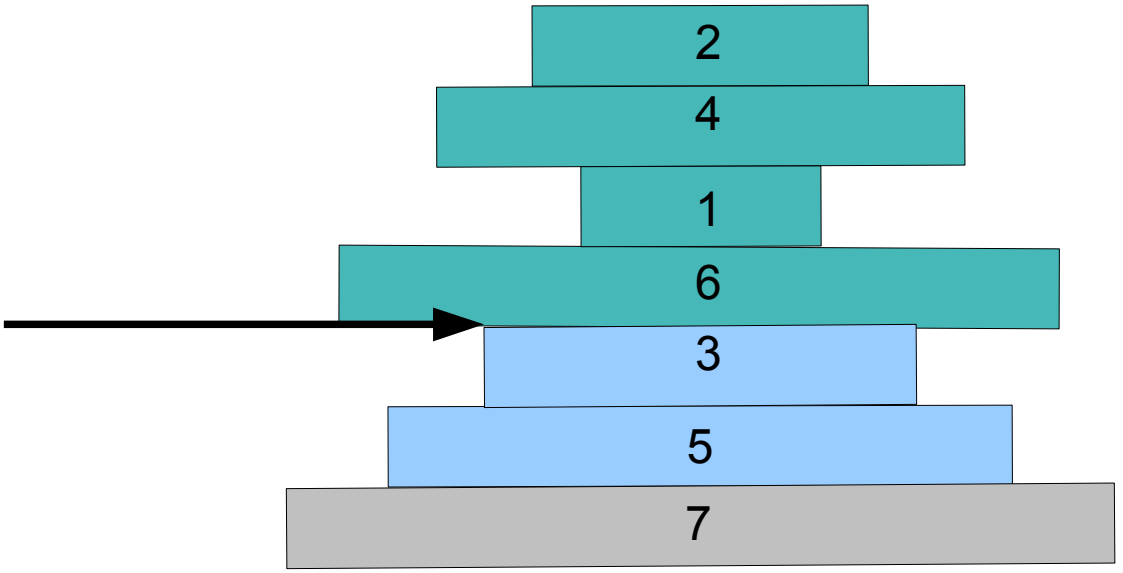
6-flip !

Pancake 5 is now at the bottom.

Pancake 2 is now at the top.

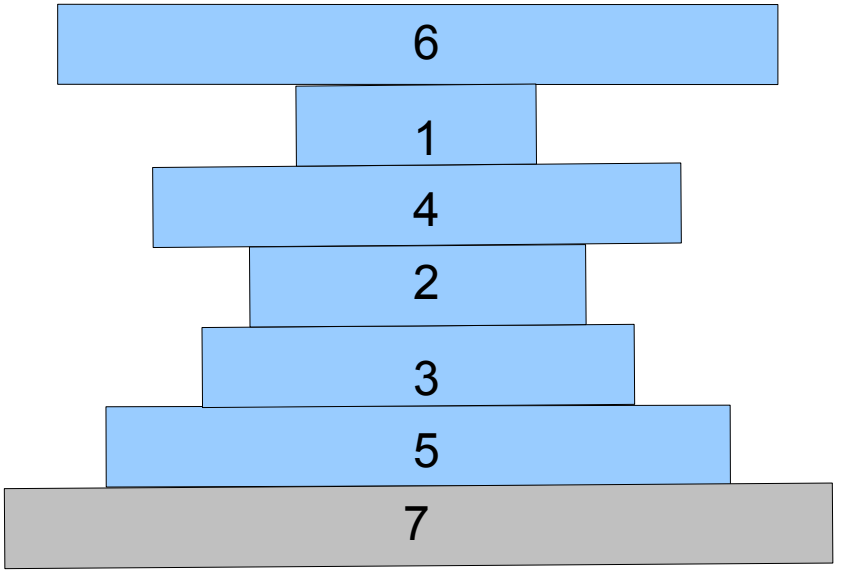


Example

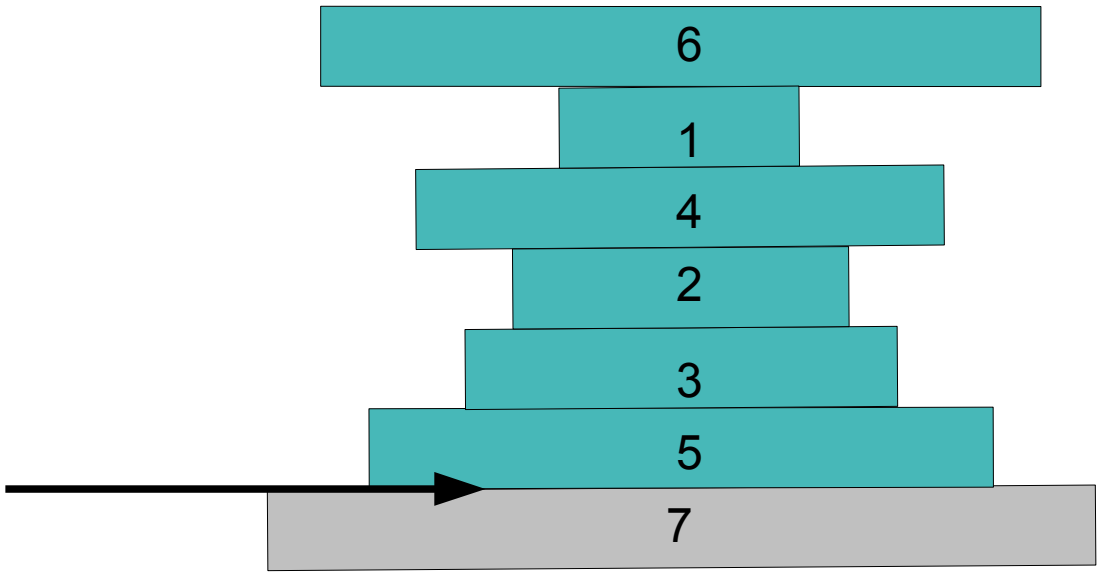


Example

4-flip !

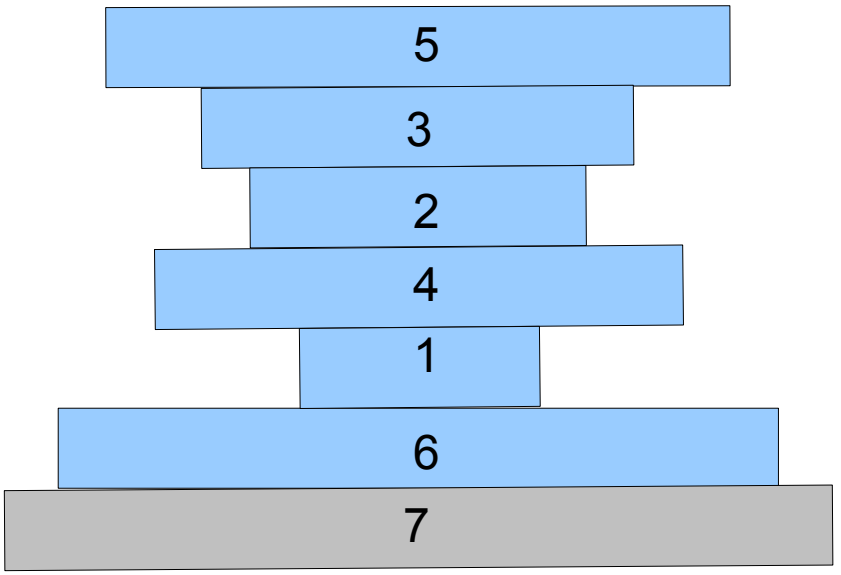


Example

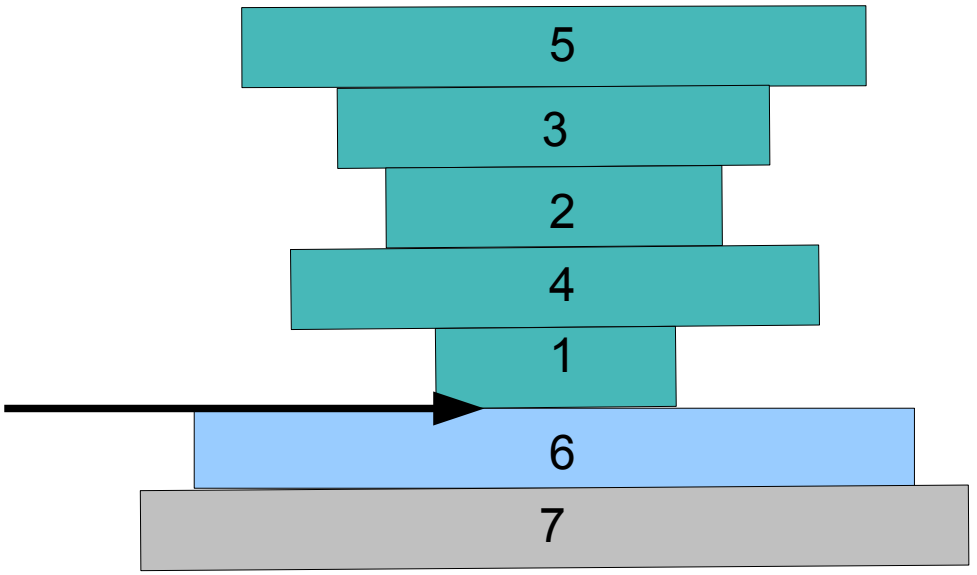


Example

6-flip !

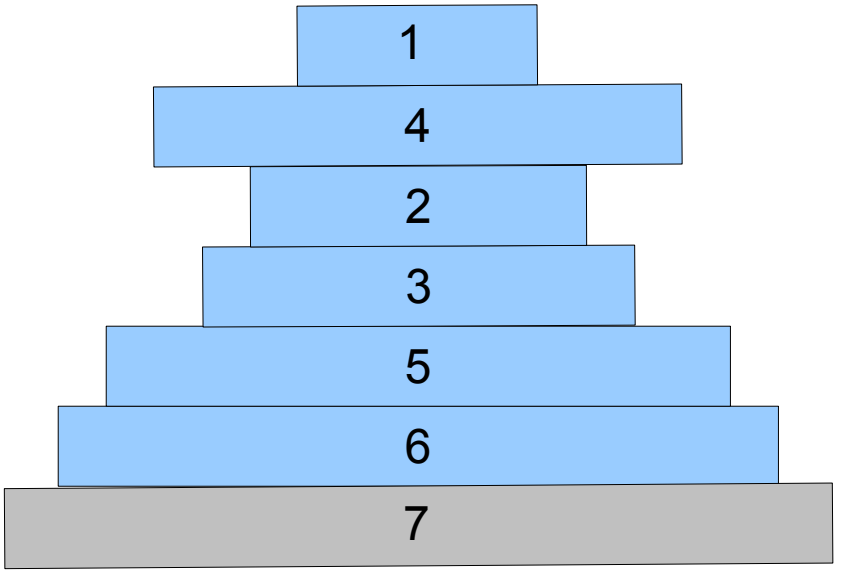


Example

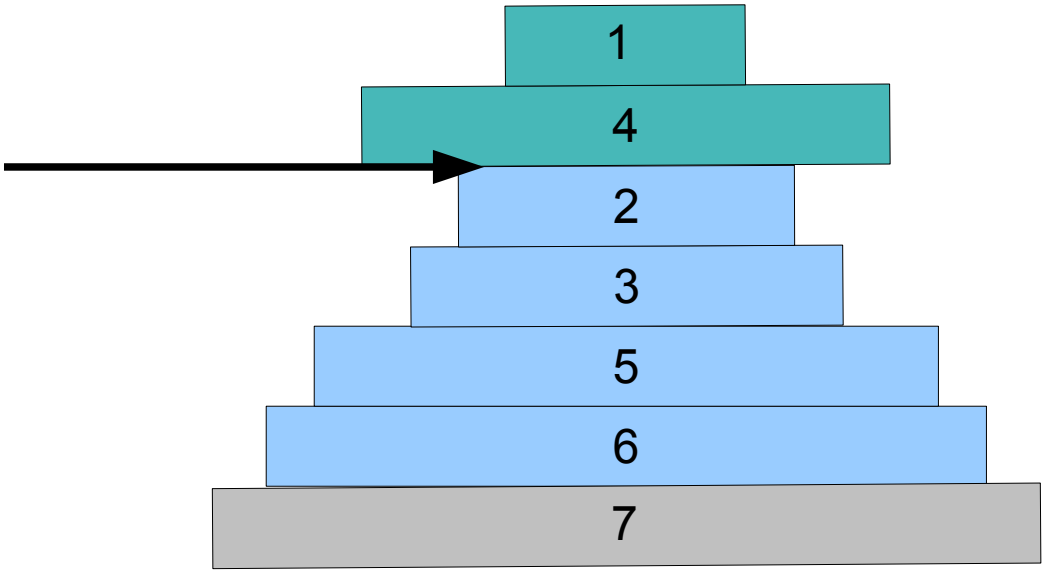


Example

5-flip !

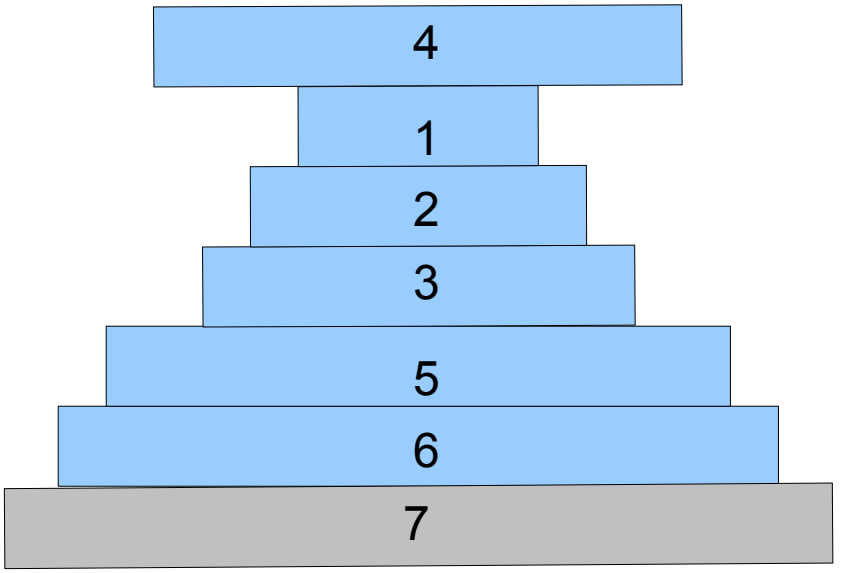


Example

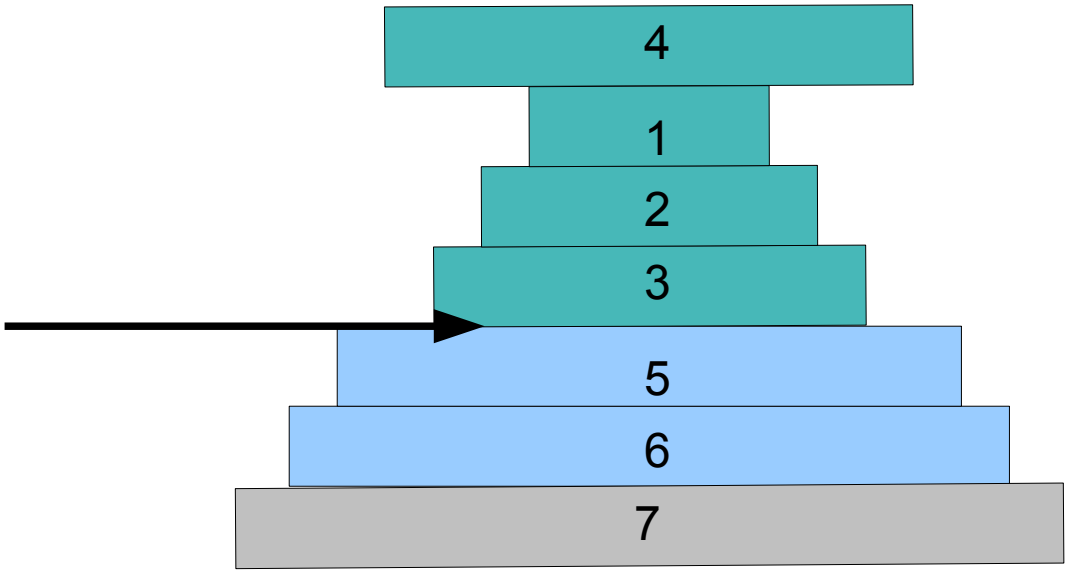


Example

2-flip !

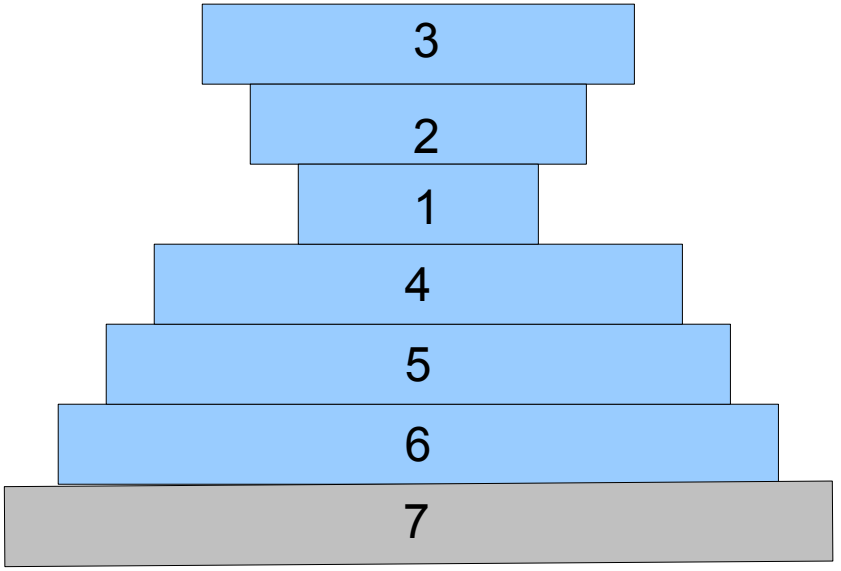


Example

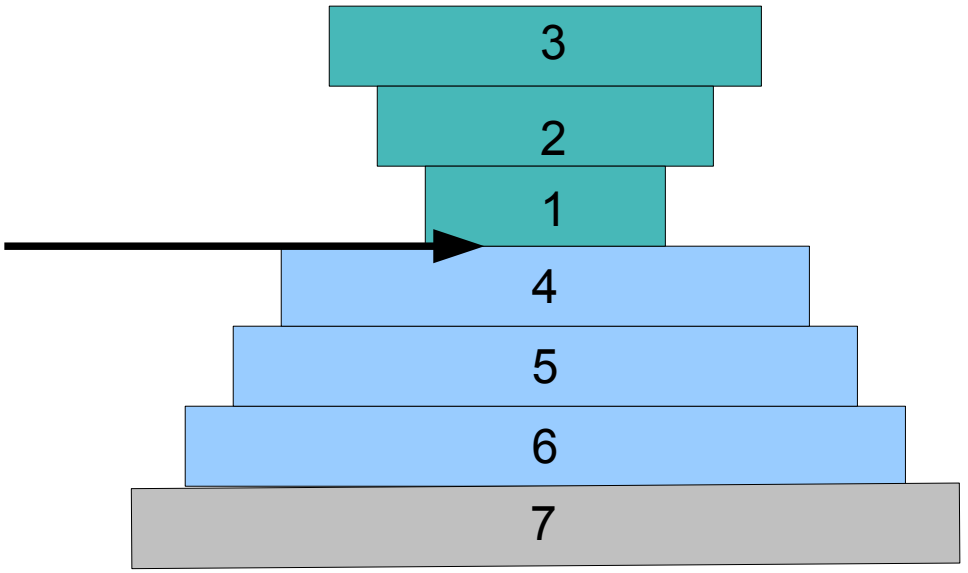


Example

4-flip !

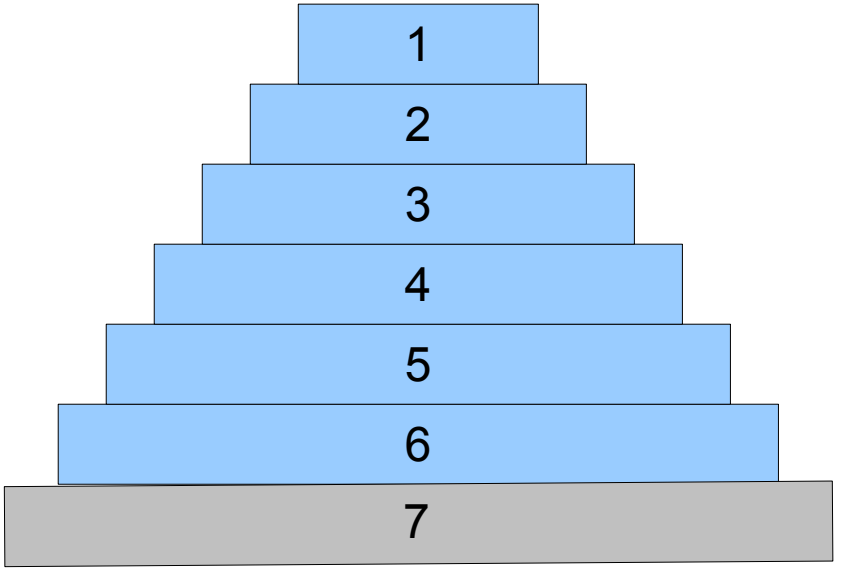


Example



Example

3-flip !



Completed in 7 moves!

3 Linked problems

- Burnt pancake problem :
  - Each pancake is burnt on one side
  - Goal : sorted stack + burnt side down.

- Genome rearrangement problem (Hayes 2007)
  - Action with two spatulas
  - Signed version : each gene has sign (+-1)
  - Unsigned version otherwise
  - Great interest in biology to classify species :

    distance(cabbage, turnip) = 3 !

4 classes of problems

#spatulas signed	1	2
yes	Burnt Pancake	Signed Genome
no	Unburnt Pancake	Unsigned Genome

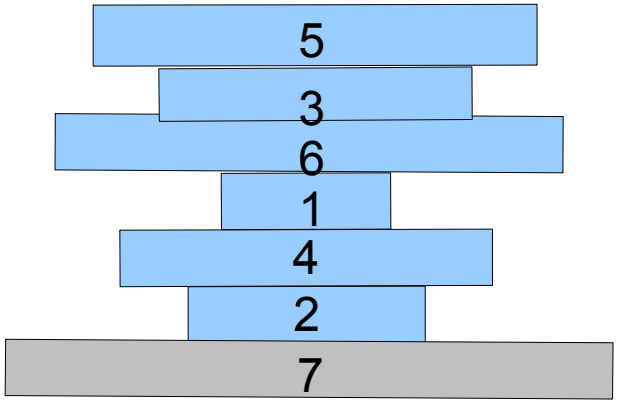
Complexities

- Unsigned genome : NP-hard (Caprara 1997)
- Signed genome : Polynomial :
  - n^4 (Pevzner & Hennenhalli 1995)
  - n^2 (Bergeron 2005)
  - n (Bader & al 2001) without the solution sequence
- Unburnt pancakes : NP-hard (Bulteau & al 2012)
- Burnt pancakes : ?
  - Polynomial on « simple » instances (Labarre & Cibulka 2011)
  - A study (Cohen & Blum 1995)
  - Another study (Cibulka 2011)

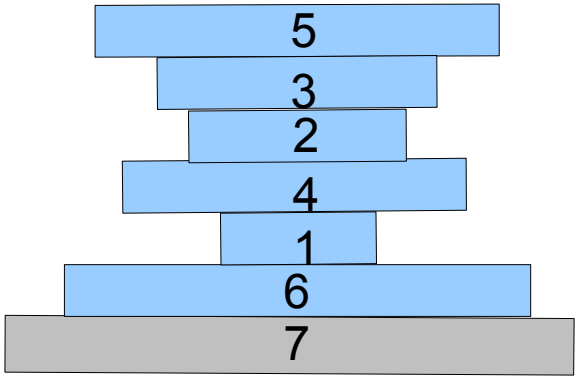
#breakpoints (#bp)

- Unsigned (unburnt) version :
  - A breakpoint is situated between two adjacent pancakes (genes) when the difference between the sizes of the two pancakes (genes) is not 1 or -1.
- Signed (burnt) version :
  - A breakpoint is situated between two adjacent pancakes (genes) when the difference between the sizes of the two pancakes (genes) is not 1.
- #breakpoints :
  - admissible heuristic for the pancake problem (Gates Papadimitriou 1979 ; Helmert 2010)
- #breakpoints/2 :
  - admissible heuristic for the genome rearrangement problem

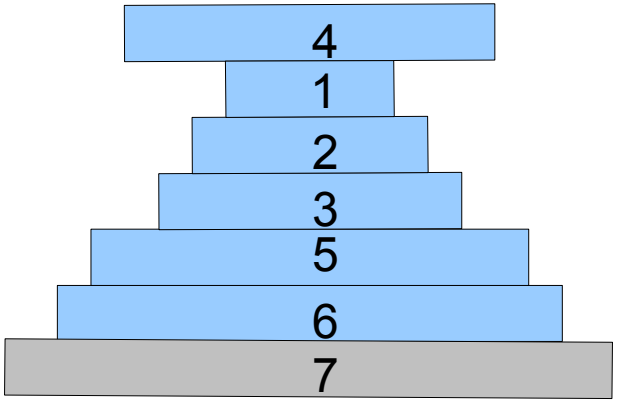
#breakpoints (#bp)



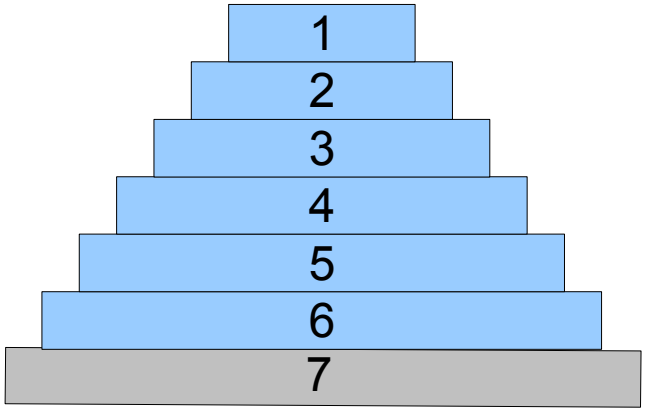
#bp=6



#bp=4



#bp=2



#bp=0

The unburnt pancake problem

- N-pancake problem
  - Stacks of size N
  - Solving a specific instance
  - Finding the diameter of the graph

- Lower and Upper bounds on the diameter of the graph
  - Gates & Papadimitriou 1995 : B- = (17/16)n
  - Heydari & Sudborough 1997 : B- = (15/14)n
  - Chitturi & al 2009 : B+ = (18/11)n

The unburnt pancake problem

- NP-hard to efficiently sort (Bulteau & al 2012)
  - At most 2 « efficient » moves : binary search.

- Efficient Sort (EffSort) (Bulteau & al 2012)
  - Only consider the « efficient » moves (that strictly decrease #bp)
  - And not the « waste » moves (that keep #bp constant)

- Alternate Sort (AS) (current work)

```
do {
    efficient sort (sequence) ; play the sequence ;
    if not completed, play a waste move ;
} while not completed ;
```

- Fixed-Depth Alternate Sort (FDAS) (current work) :

  Alternate Sort using fixed-depth EffSort

The unburnt pancake problem

- The pancake challenge (2004)
  - Tomas Rokicki entry
    - Backward Search : the reverse sequence of a solution to Pi is a solution to Pi^-1
    - Building sequences hard to solve.
    - Other clever tricks

- 2-approximation algorithm (Fischer & Ginzinger 2005)
  - Type 1, 2, 3, 4 moves
  - R-approx = L / #bp
    - <= 2 (theory)
    - == 1.22 on average (practice)

Two general tools

- Iterative Deepening Depth-First Search (IDDFS) (Korf 1985)
  - Needs an Heuristic = #breakpoints (Helmert 2010)
  - Solves random instances of size 60 or 80 in a few seconds on average
  - Cannot solve some specific complex stacks of size 30.
  - Time to solve size N stacks : exponential in N
  - While not completed, a lower bound is provided
  - Optimal solution at completion
- (Nested) Monte-Carlo Search ((N)MCS) (Cazenave 2009)
  - Needs domain-dependant simulations
  - Solves random instances of size up to 512.
  - The time to solve a stack is polynomial in N
  - While not completed, an upper bound is provided
  - Stops only if the optimal length == heuristic value
  - Otherwise, no way to see if the solution is optimal

Domain-dependant tools

- Simulators
  - Fixed-Depth Alternate Sort
  - Fischer and Ginzinger (FG) algorithm
  - BREF

```
while not completed {
    choose an efficient move at random ;
    If (success) play it ;
    else { choose a waste move at random ; play it ; }
}
```

- Backward Solutions (BS) (Rokicki 2004)

Settings of experiments

- For a specific problem, finding an optimal solution.

- For a set of X problems drawn at random
  - R-approx = L / #breakpoints
  - Less than one hour to solve a set a problems
  - Standard deviation of r-approx onone problem = 0.05

    ==> 2-sigma rule : precision of (0.05 2 / 10 =) 0.01 on r-approx averaged on 100 problems

Results of experiments (1)

- IDDFS is optimal and R > 1 ? (normal because #bp is a lower bound)
- IDDFS cannot solve instances of size > 64
- MCS can solve instances of sizes up to 256
- For N<= 64, MCS is inferior to IDDFS

		IDDFS			NMCS	+ BREF	
N	L	R	T	L	R	T	level
8	6.5	1.09	0	6.5	1.09	0	4
16	14.5	1.05	0	15.5	1.09	0.05	4
32	31.0	1.03	0	36.5	1.21	0.6	3
64	63.0	1.02	5	76	1.24	1.2	2
128	_	_	∞	163	1.29	0.3	1
256	_	_	∞	333	1.31	9	1

Results of experiments (2)

- BS enhancement : l = min(l1, l2)

  l1 = length of the forward simulation

  l2 = length of the backward simulation on the inverse problem

- Left : MCS + BREF
- Right : MCS + BREF + BS

		NMCS	+BREF			NMCS	+BREF	+ BS
N	L	R	T	level	L	R	T	level
8	6.5	1.09	0	4	6.6	1.10	0	4
16	15.5	1.09	0.05	4	15.3	1.09	0.1	4
32	36.5	1.21	0.6	3	35.2	1.15	0.8	3
64	76	1.24	1.2	2	75.2	1.20	2	2
128	163	1.29	0.3	1	159	1.26	0.5	1
256	333	1.31	9	1	335	1.32	0.1	0

Results of experiments (3)

- Fischer and Ginzinger (FG)
- FG simulation (R-approx=2 in theory and 1.22 in practice)
- Left : MCS + BREF + BS
- Right : MCS + FG + BS

		NMCS	+BREF	+ BS		NMCS	+FG	+ BS
N	L	R	T	level	L	R	T	level
8	6.6	1.10	0	4	6.7	1.10	0	4
16	15.3	1.09	0.1	4	14.8	1.05	0.6	4
32	35.2	1.15	0.8	3	31.6	1.05	4.2	3
64	75.2	1.20	2	2	68.5	1.10	2.5	2
128	159	1.26	0.5	1	155	1.23	0.8	1
256	335	1.32	0.1	0	336	1.32	0.2	0

Results of experiments (4)

- Fixed-depth Alternate Sort (FDAS)
- Left : MCS + FG + BS
- Right : MCS + FDAS
- Solving size 512 stacks

		NMCS	+FG	+ BS		NMCS	+ FDAS	
N	L	R	T	level	L	R	T	level
8	6.7	1.10	0	4	6.6	1.09	0	3
16	14.8	1.05	0.6	4	14.5	1.04	0.04	3
32	31.6	1.05	4.2	3	31.3	1.04	0.8	3
64	68.5	1.10	2.5	2	65.1	1.04	18	3
128	155	1.23	0.8	1	139	1.10	10.5	2
256	336	1.32	0.2	0	300	1.18	1.5	1
512	_	_	_	_	614	1.20	9	1

Conclusions

- The (unburnt) pancake problem is experimentally revisited.
- Associated with :
  - BREF, FG, FDAS as simulators,
- MCS extends the results obtained by IDDFS
  - Sizes of stacks up to 512 with r-approx=1.20
  - Rapprox = 1.4 for N in[16, 64]
  - Rapprox in [1.10, 1.20] for N in [128, 512]

Future work

- The burnt pancake problem
  - IDDFS : Improve the Cibulka's heuristic function
  - MCS : assess Cohen & Blum algorithm as simulator
  - Find new results on the diameter

- The unburnt pancake problem
  - Find new results on the diameter
  - Generate complex stacks

Thank you for your attention!