

PROBLEM SOLVING

[BACK](#)

Pólya's approach to problem-solving



image source: <http://doi.org/10.3932/ethz-a-000099441>

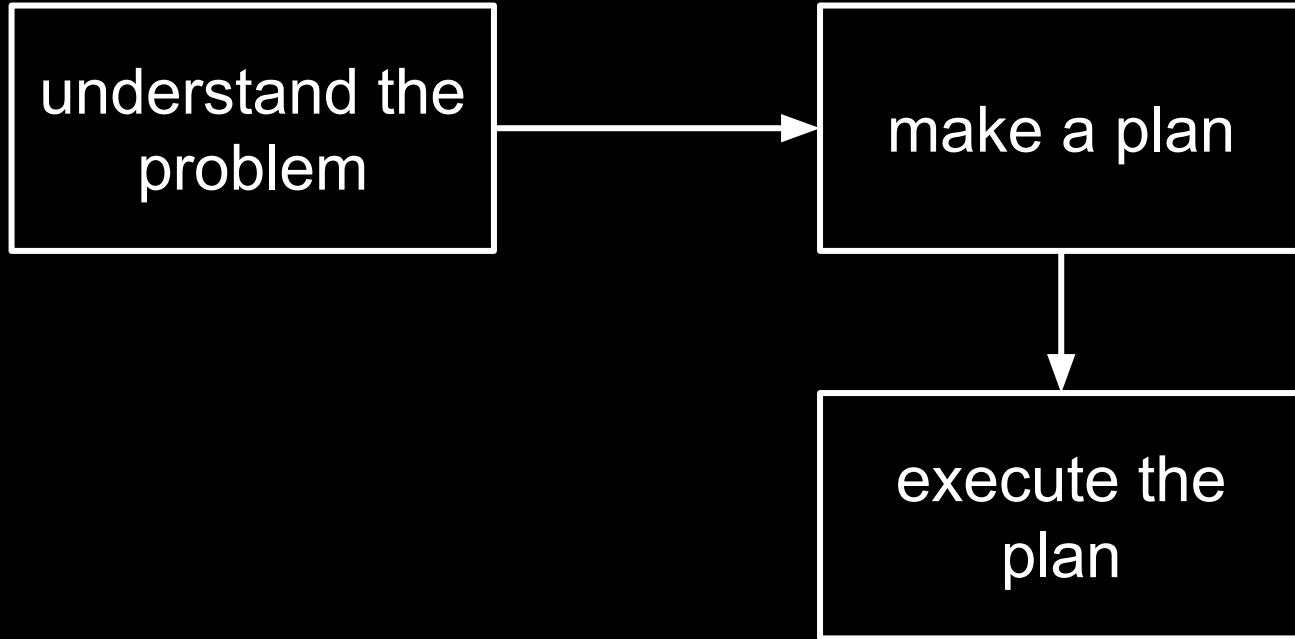
Pólya's approach to problem-solving

understand the
problem

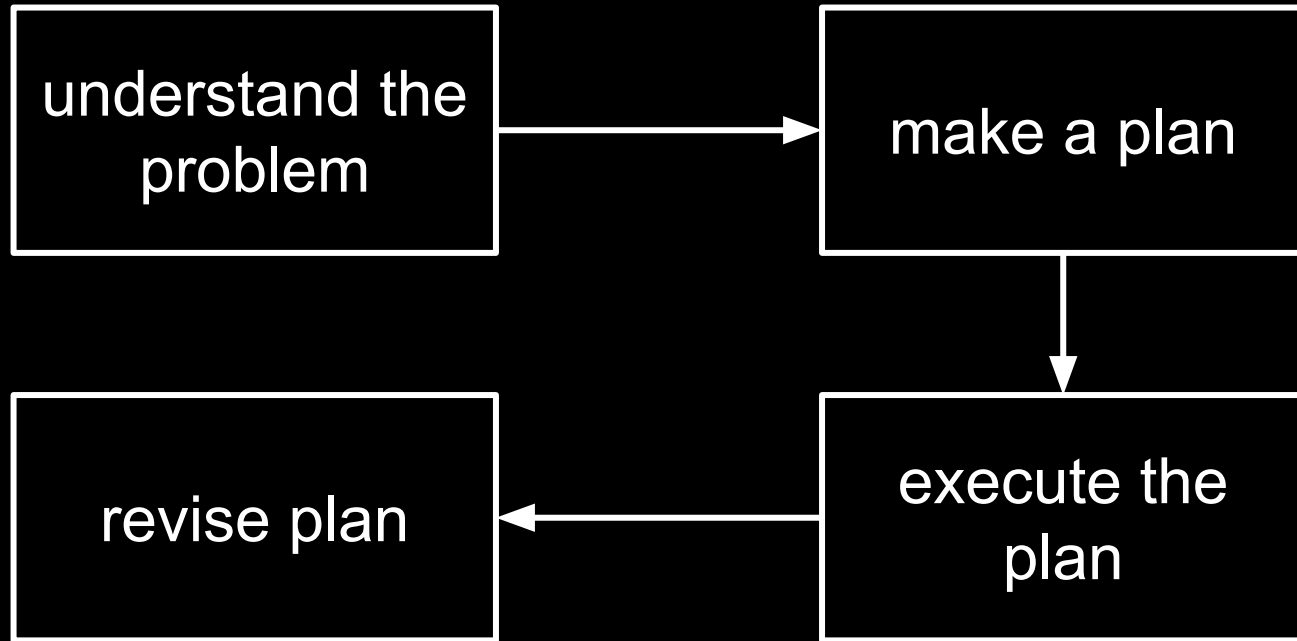
Pólya's approach to problem-solving



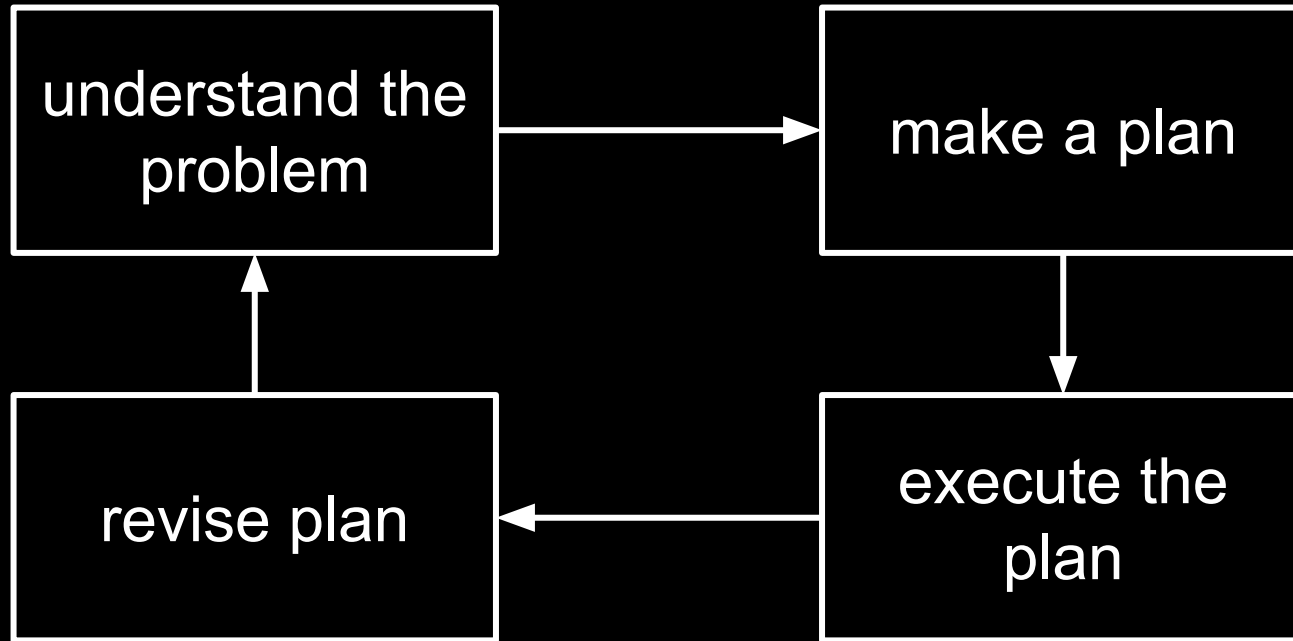
Pólya's approach to problem-solving



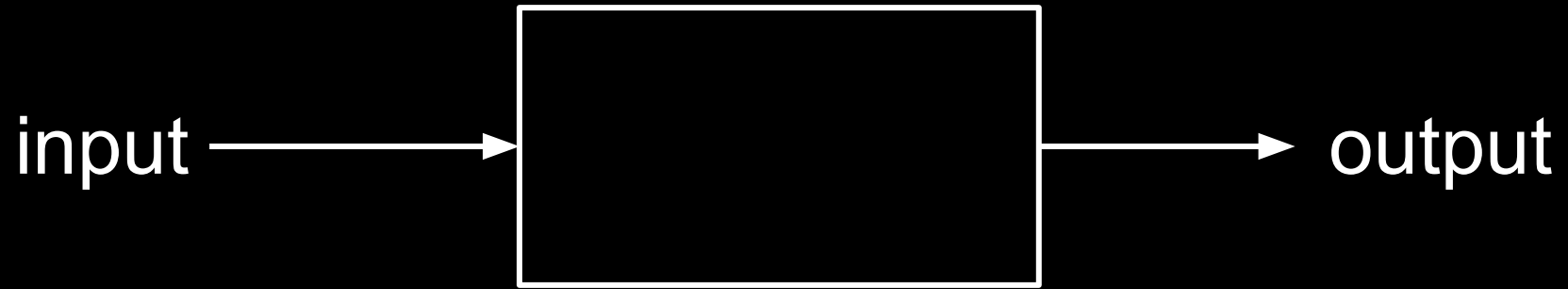
Pólya's approach to problem-solving



Pólya's approach to problem-solving



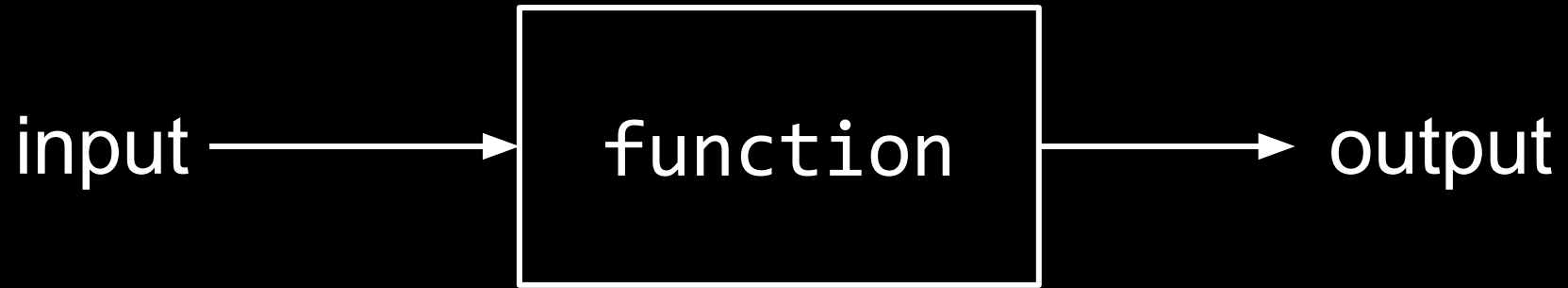
a model to represent problems



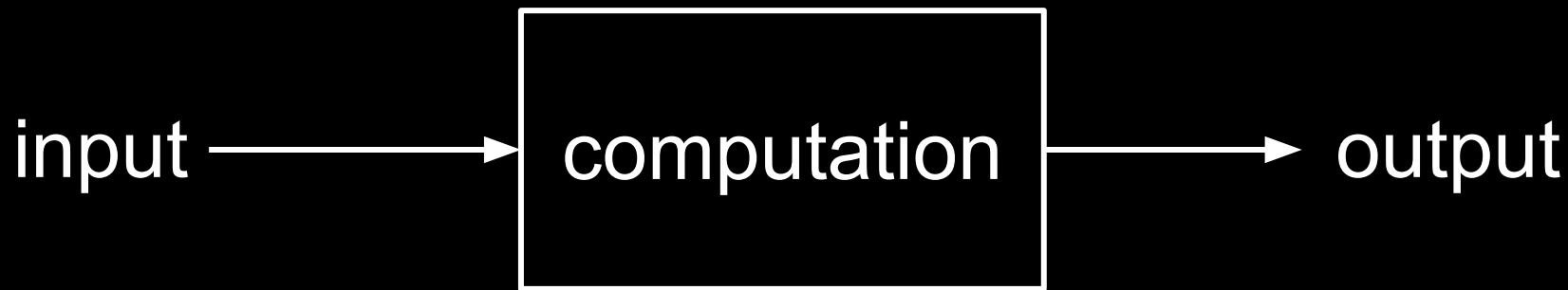
a model to represent problems



a model to represent problems

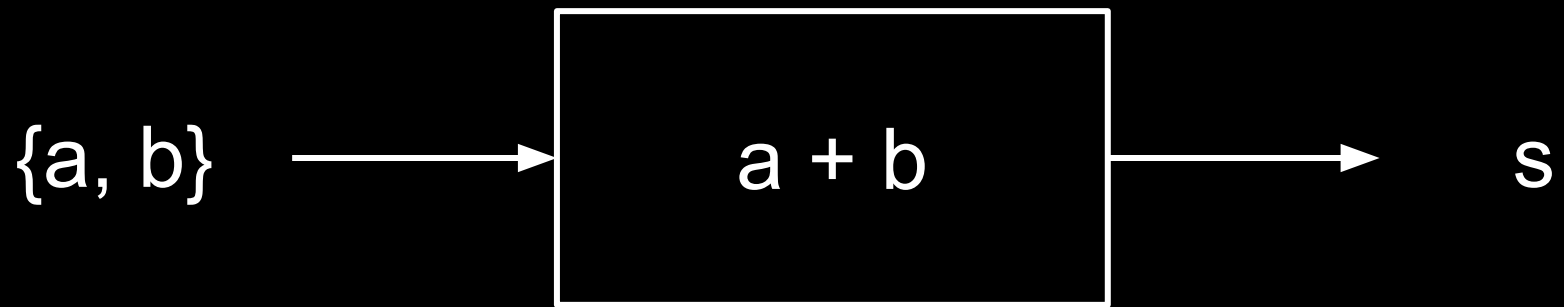


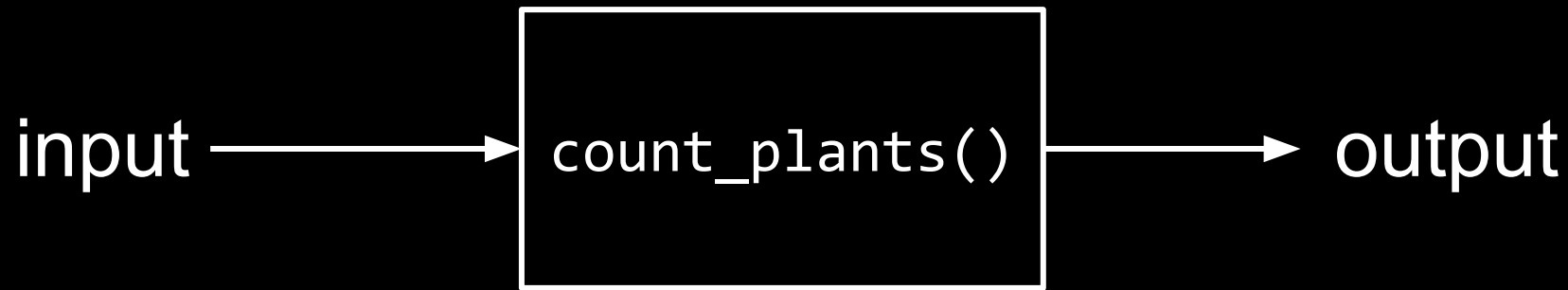
a model to represent problems

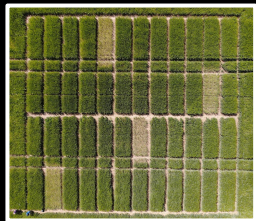


the rainbow experiment as an
input - processing - output - problem

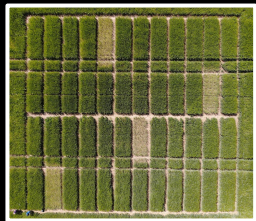






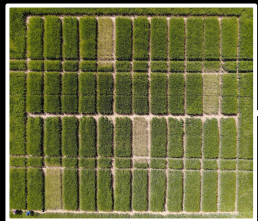


output



42

processing of
information



`count_plants()`

42

representation of
information



next_move()

E2 → E4



1: 0R	9: 0P	57: 1R
2: 0N	10: 0P	58: 1N
3: 0B	11: 0P	59: 1B
4: 0K	12: 0P	60: 1K
5: 0Q	13: 0P	61: 1Q
6: 0B	14: 0P	62: 1B
7: 0N	15: 0P	63: 1N
8: 0R	16: 0P	64: 1R

...

representation of information



problem solving strategies

problem decomposition

large and complex problem

less complex
subproblem

less complex
subproblem

less complex subproblem

less complex
subproblem

less complex
subproblem

less complex subproblem

less complex subproblem

less complex subproblem

divide and conquer

large and complex problem of type A

smaller problem
of type A

smaller problem
of type A

smaller problem
of type A

smaller problem
of type A

even smaller problem of type A	even smaller problem of type A
even smaller problem of type A	even smaller problem of type A
even smaller problem of type A	even smaller problem of type A
even smaller problem of type A	even smaller problem of type A

sorted list +
element



search()



yes / no

is 67 a prime number?

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

linear search



2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

linear search



2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

linear search



2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

linear search



2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

linear search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
~~43~~, ~~47~~, ~~53~~, ~~59~~, ~~61~~, 67, 71, 73, 79, 83, 89, 97
↑

19 steps... can't we do better?

2, 3, 5, 7, 11, ~~13~~, 17, ~~19~~, ~~23~~, ~~29~~, ~~31~~, ~~37~~, ~~41~~,
~~43~~, ~~47~~, ~~53~~, ~~59~~, ~~61~~, 67, 71, 73, 79, 83, 89, 97

↑

large and complex
problem

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

large and complex
problem

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

smaller
problem

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41

smaller
problem

43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

binary search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

binary search

67 != 41



2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ~~41~~,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

binary search

67 > 41



2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, ~~41~~,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

binary search

67 > 41



2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97

binary search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97



67 != 71

binary search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97



67 != 71

binary search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97



$67 < 71$

binary search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97



67 != 59

binary search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
~~43~~, 47, ~~53~~, 59, 61, 67, ~~71~~, ~~73~~, 79, ~~83~~, ~~89~~, 97



67 > 59

binary search

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41,
~~43~~, ~~47~~, ~~53~~, ~~59~~, 61, 67, ~~71~~, ~~73~~, ~~79~~, ~~83~~, ~~89~~, 97



67 = 67

binary search

2, 3, 5, 7, 11, ~~13~~, 17, 19, ~~23~~, ~~29~~, 31, 37, 41,
~~43~~, 47, ~~53~~, 59, ~~61~~, 67, ~~71~~, ~~73~~, 79, ~~83~~, ~~89~~, 97



67 = 67

3 splits → much better

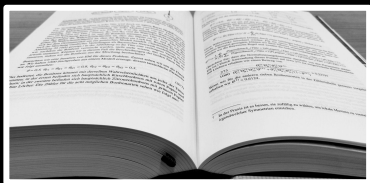
2, 3, 5, 7, 11, ~~13~~, 17, 19, ~~23~~, 29, 31, 37, 41,
~~43~~, 47, ~~53~~, 59, ~~61~~, 67, ~~71~~, ~~73~~, 79, ~~83~~, ~~89~~, 97



67 = 67



how efficient are linear and
binary search in general?



`count_words()`

word count

Die Verteilung der Bonbons in jeder Tüte wird durch ein solches Bayes'sches Modell beschrieben, das eine Wahrscheinlichkeitsverteilung für jedes Merkmal (z.B. die Farbe) und die Wahrscheinlichkeit, dass ein Bonbon einer bestimmten Farbe ist, enthält. Die Parameter θ_1 und θ_2 sind die Wahrscheinlichkeiten, dass ein Bonbon eine bestimmte Farbe hat, und die Bonbons stimmen mit Tüte 1 überein. Das Ziel besteht darin, die Wahrscheinlichkeiten an, dass das Bonbon ein Loch hat, zu schätzen, indem wir eine Iteration von EM für dieses Problem. Zuerst sehen wir uns die Daten an. Wir haben 1000 Stichproben aus einem Modell erzeugt, dessen tatsächliche Parameter $\theta = 0.5$, $\theta_1 = \theta_{r1} = \theta_{r2} = 0.8$, $\theta_2 = \theta_{w2} = \theta_{l2} = 0.3$.

Dies bedeutet, die Bonbons können mit derselben Wahrscheinlichkeit aus jeder der Tüten kommen; in der ersten befinden sich hauptsächlich Kirschbonbons mit rotem Papier und in der zweiten befinden sich hauptsächlich Zitronenbonbons mit grünem Papier ohne Löcher. Die Zähler für die acht möglichen Bonbonarten sehen wie folgt aus:

(20.7)

Wenn wir für die anderen sieben Bonbonsorten in der Zählentabelle genauso vorgehen erhalten wir $\theta^{(1)} = 0.6124$.

In der Praxis ist es besser, sie zufällig zu wählen, um lokale Maxima zu vermeiden, irgendwelcher Symmetrien entstehen.

Die Verteilung der Bonbons in jeder Tüte wird durch ein solches Bayes'sches Modell beschrieben, das eine Wahrscheinlichkeitsverteilung für jedes Merkmal (z.B. die Farbe) und die Wahrscheinlichkeit, dass ein Bonbon einer dieser Merkmale zugeordnet wird, enthält. Die Parameter θ_1 und θ_2 sind die Wahrscheinlichkeiten, dass ein Bonbon einer dieser Merkmale zugeordnet wird, und die Bonbons stimmen mit Tüte 1 bzw. Tüte 2 überein. Das Ziel besteht darin, die Wahrscheinlichkeiten an, dass das Bonbon ein Loch hat, zu schätzen, indem wir eine Iteration von EM für dieses Problem. Zuerst sehen wir uns die Daten an. Wir haben 1000 Stichproben aus einem Modell erzeugt, dessen tatsächliche Parameter $\theta = 0.5$, $\theta_1 = \theta_{r1} = \theta_{r2} = 0.8$, $\theta_2 = \theta_{w1} = \theta_{w2} = 0.3$.

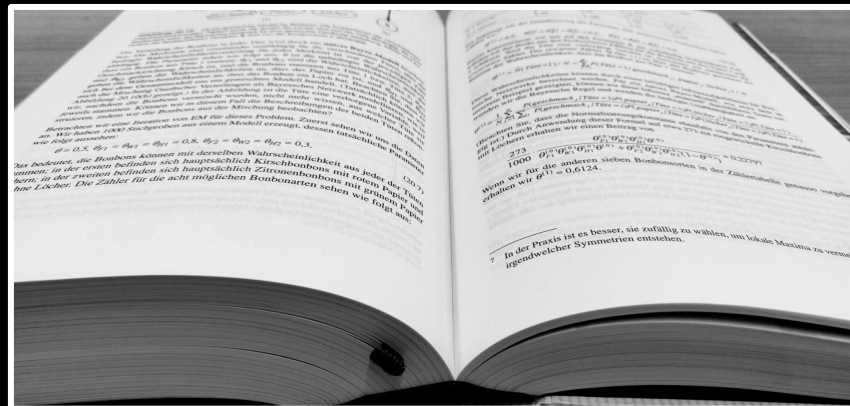
Dies bedeutet, die Bonbons können mit derselben Wahrscheinlichkeit aus jeder der Tüten kommen; in der ersten befinden sich hauptsächlich Kirschbonbons mit rotem Papier und in der zweiten befinden sich hauptsächlich Zitronenbonbons mit grünem Papier ohne Löcher. Die Zähler für die acht möglichen Bonbonarten sehen wie folgt aus:

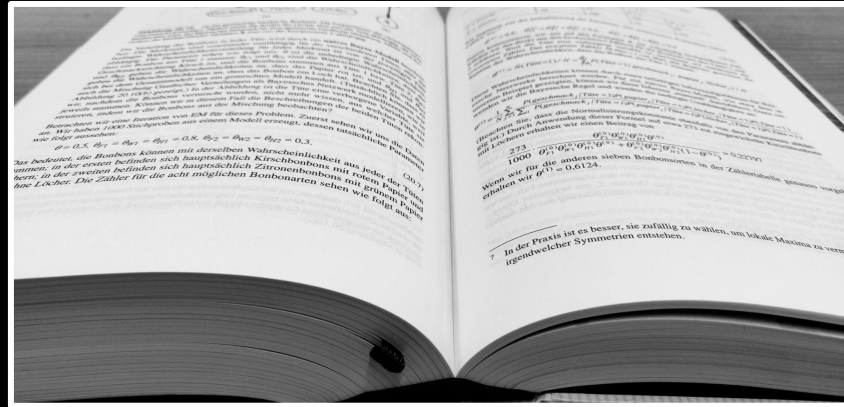
(20.7)

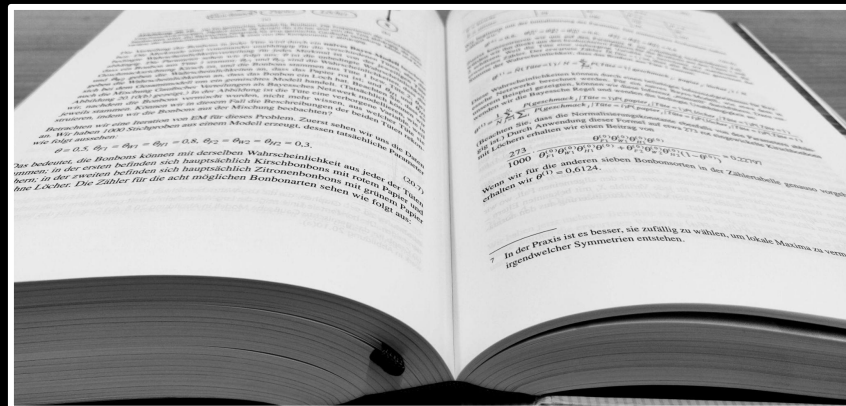
Wenn wir für die anderen sieben Bonbonsorten in der Zählentabelle genauso vorgehen erhalten wir $\theta^{(1)} = 0.6124$.

In der Praxis ist es besser, sie zufällig zu wählen, um lokale Maxima zu vermeiden, irgendwelcher Symmetrien entstehen.

strategies, anyone?



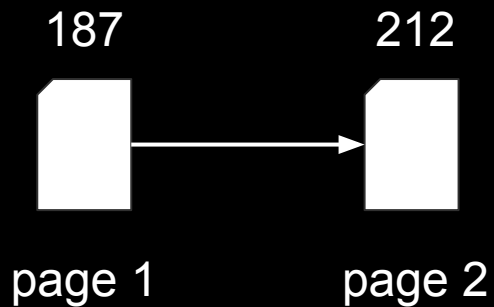
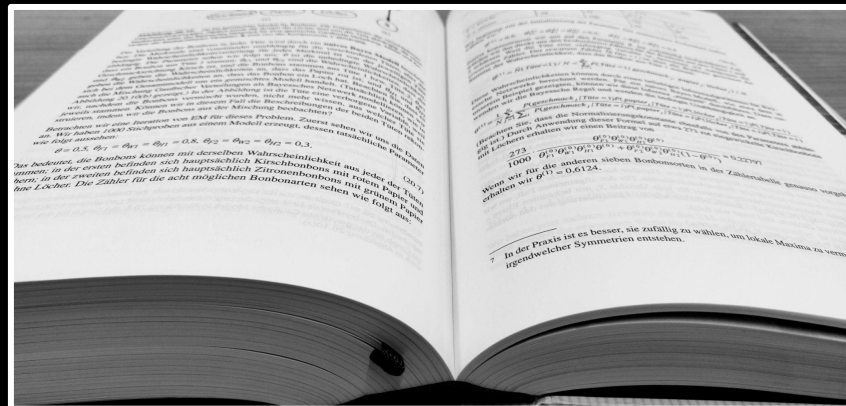


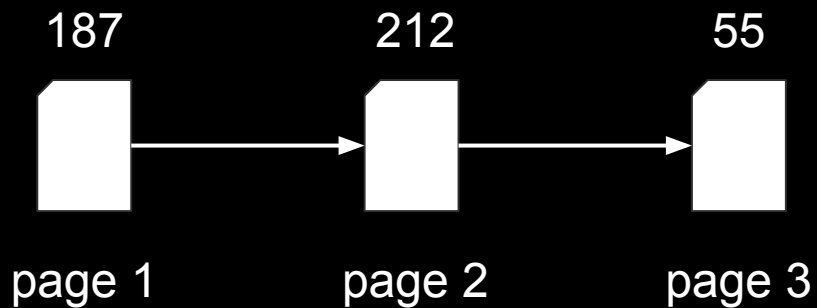
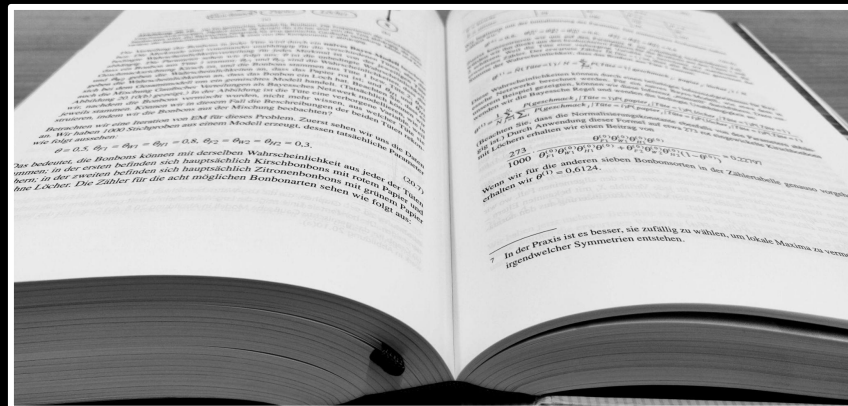


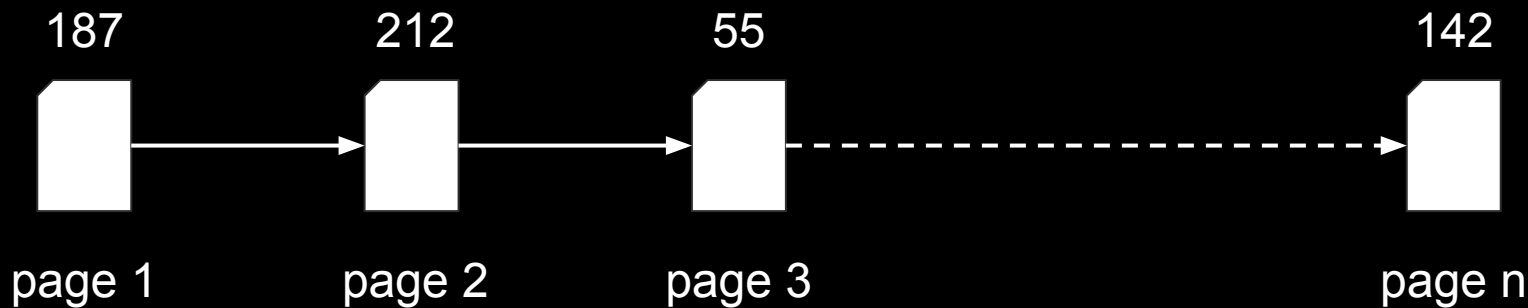
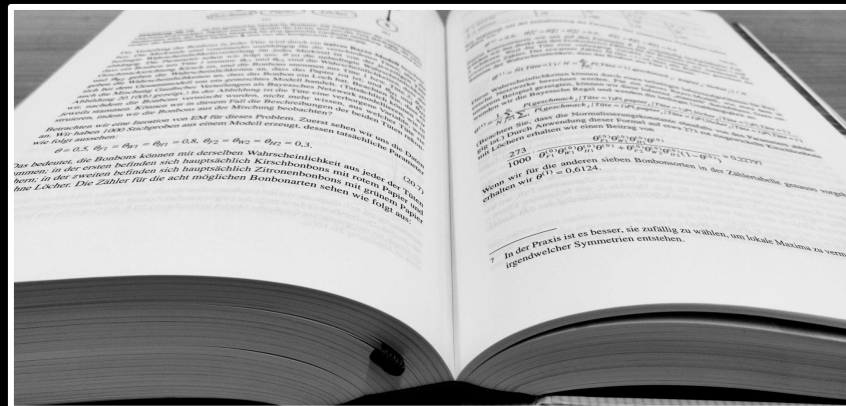
187

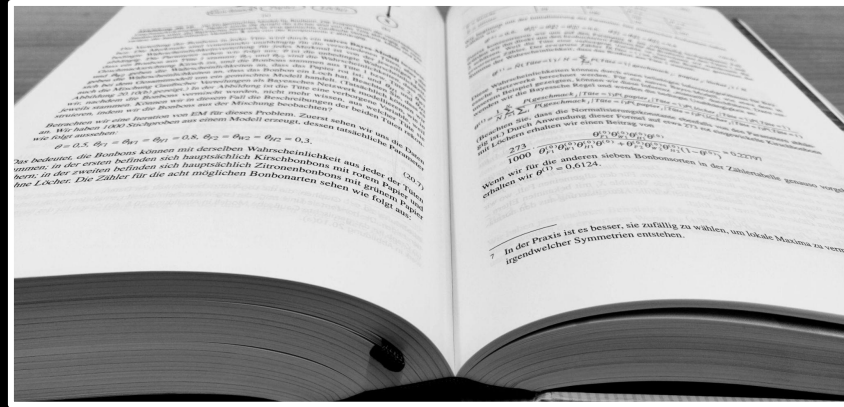


page 1









n = 1327 pages

Ø 2:23 minutes per page

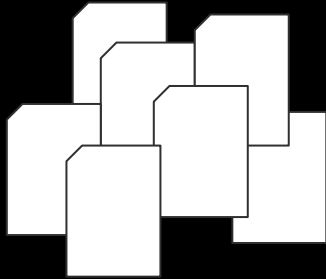
~ 52.34 hours

divide and conquer

+

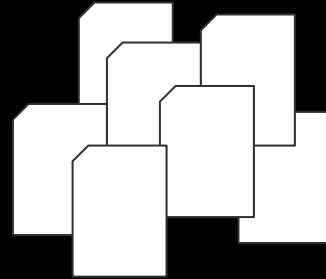
?

pages 1 - 700



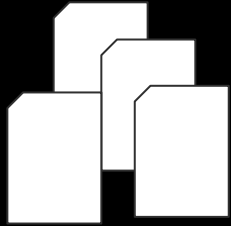
student 1

pages 701 - 1327



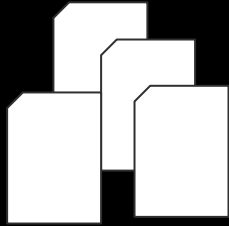
student 2

pages 1 - 350



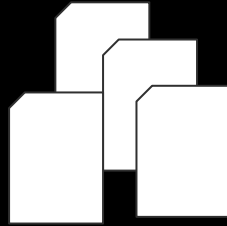
student 1

pages 351 - 700



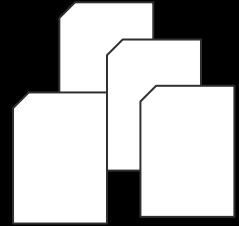
student 2

pages 701 - 1050



student 3

pages 1051- 1327



student 4

divide and conquer

+

distribution and parallelization

