Informatik II

Exercise 1

Andreas Bärtschi, Andreea Ciuprina, Felix Friedrich, Patrick Gruntz, Hermann Lehner, Max Rossmannek, Chris Wendler

FS 2018

Program for Today

1. Administrative
2. Repetition Theory
   - Problem and Algorithm
   - Aysmptotic Running Time
3. Programming Exercise
   - Toss an Unfair Dice

Offer

- Doing the weekly exercise series $\rightarrow$ bonus of maximally 0.25 of a grade points for the exam.
- The bonus is proportional to the achieved points of **specially marked bonus-task**. The full number of points corresponds to a bonus of 0.25 of a grade point.
- For the **admission** to bonus task 1 you need to gain 180 points on the first three exercise tasks.
- Rationale: You should have had a serious look at the exercise before doing the bonus task.
- The bonus task is unlocked as soon as you have the required 180 points, but not earlier than to the third week.

Warm-up

- What is a problem?

Warm-up

- What is a problem?
- What is an algorithm?

Warm-up

- What is a problem?
- What is an algorithm?
  - ➜ well-defined computing procedure to compute output data from input data.

Warm-up

- What is a problem?
- What is an algorithm?
  - ➜ well-defined computing procedure to compute output data from input data.
- What is a program?

Warm-up

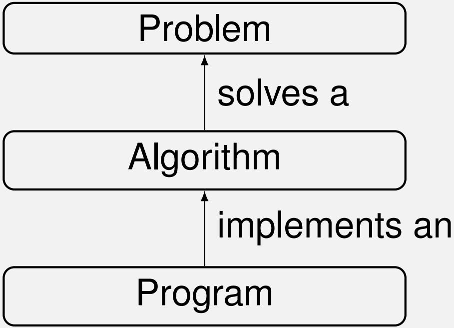
- What is a problem?
- What is an algorithm?

  → well-defined computing procedure to compute output data from input data.

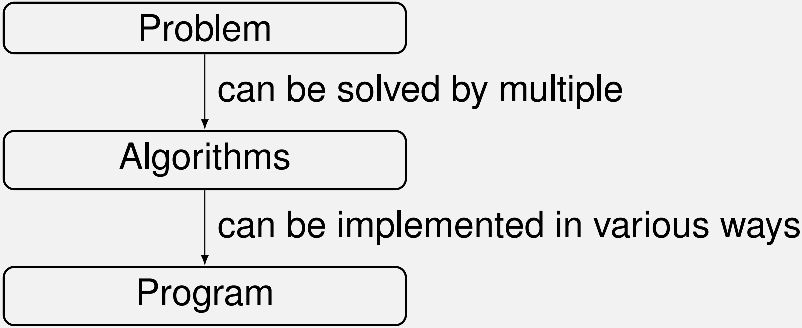
- What is a program?

  → Concrete implementation of an algorithm

Warm-up



Warm-up


Problem
can be solved by multiple
Algorithms
can be implemented in various ways
Program

Efficiency

Problem	Complexity	Minimal (asymptotic) cost over all algorithms that solve the problem.
Algorithm	Cost	Number of elementary operations
Program	Computing time	Measurable value on an actual machine.

Efficiency

Problem	Complexity	Minimal (asymptotic) cost over all algorithms that solve the problem.
Algorithm	Cost	Number of elementary operations
Program	Computing time	Measurable value on an actual machine.

➜ Estimation of cost or computing time depending on the input size, denoted by n.

Asymptotic behavior

- What are $\Omega(g(n))$, $\Theta(g(n))$, $\mathcal{O}(g(n))$?

Asymptotic behavior

- What are $\Omega(g(n))$, $\Theta(g(n))$, $\mathcal{O}(g(n))$?

➜ Sets of functions!

Asymptotic behavior

- What are $\Omega(g(n))$, $\Theta(g(n))$, $\mathcal{O}(g(n))$?

→ Sets of functions!

Repetition, sets A, B:

subset	$A \subseteq B$
proper subset	$A \subsetneq B$
intersection	$A \cap B$

Asymptotic behavior

Given: function $f : \mathbb{N} \to \mathbb{R}$.

Definition:

$$\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{R} | \exists c > 0, n_0 \in \mathbb{N} : 0 \leq f(n) \leq c \cdot g(n) \; \forall n \geq n_0\}$$
$$\Omega(g) = \{f : \mathbb{N} \to \mathbb{R} | \exists c > 0, n_0 \in \mathbb{N} : 0 \leq c \cdot g(n) \leq f(n) \; \forall n \geq n_0\}$$
$$\Theta(g) = \mathcal{O}(g) \cap \Omega(g)$$

Useful information for the exercise

Theorem

1. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f \in \mathcal{O}(g), \mathcal{O}(f) \subsetneq \mathcal{O}(g)$.
2. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = C > 0$ *(C constant)* $\Rightarrow f \in \Theta(g)$.
3. $\frac{f(n)}{g(n)} \underset{n\to\infty}{\longrightarrow} \infty \Rightarrow g \in \mathcal{O}(f), \mathcal{O}(g) \subsetneq \mathcal{O}(f)$.

Useful information for the exercise

Theorem

1. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f \in \mathcal{O}(g), \mathcal{O}(f) \subsetneq \mathcal{O}(g)$.
2. $\lim_{n\to\infty} \frac{f(n)}{g(n)} = C > 0$ *(C constant)* $\Rightarrow f \in \Theta(g)$.
3. $\frac{f(n)}{g(n)} \underset{n\to\infty}{\longrightarrow} \infty \Rightarrow g \in \mathcal{O}(f), \mathcal{O}(g) \subsetneq \mathcal{O}(f)$.

Beispiel

1. $\lim_{n\to\infty} \frac{n}{n^2} = 0 \Rightarrow n \in \mathcal{O}(n^2), \mathcal{O}(n) \subsetneq \mathcal{O}(n^2)$.
2. $\lim_{n\to\infty} \frac{2n}{n} = 2 > 0 \Rightarrow 2n \in \Theta(n)$.
3. $\frac{n^2}{n} \underset{n\to\infty}{\longrightarrow} \infty \Rightarrow n \in \mathcal{O}(n^2), \mathcal{O}(n) \subsetneq \mathcal{O}(n^2)$.

Quiz

$1 \in \mathcal{O}(15)$?

Quiz

$1 \in \mathcal{O}(15)$? ✓ better $1 \in \mathcal{O}(1)$

Quiz

$1 \in \mathcal{O}(15)$? ✓ better $1 \in \mathcal{O}(1)$

$2n + 1 \in \Theta(n)$?

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓
$\sqrt{n} \notin \Theta(n)$?	

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n+1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓
$\sqrt{n} \notin \Theta(n)$?	✓

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓
$\sqrt{n} \notin \Theta(n)$?	✓
$\mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n)$?	

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓
$\sqrt{n} \notin \Theta(n)$?	✓
$\mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n)$?	✓

Quiz

$1 \in \mathcal{O}(15)$	?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$	?	✓
$\sqrt{n} \in \mathcal{O}(n)$	?	✓
$\sqrt{n} \in \Omega(n)$	?	✗
$n \in \Omega(\sqrt{n})$	?	✓
$\sqrt{n} \notin \Theta(n)$	?	✓
$\mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n)$	?	✓
$2^n \notin \mathcal{O}(\exp(n))$	?	

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n+1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓
$\sqrt{n} \notin \Theta(n)$?	✓
$\mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n)$?	✓
$2^n \notin \mathcal{O}(\exp(n))$?	✗

Quiz

$1 \in \mathcal{O}(15)$?	✓ better $1 \in \mathcal{O}(1)$
$2n + 1 \in \Theta(n)$?	✓
$\sqrt{n} \in \mathcal{O}(n)$?	✓
$\sqrt{n} \in \Omega(n)$?	✗
$n \in \Omega(\sqrt{n})$?	✓
$\sqrt{n} \notin \Theta(n)$?	✓
$\mathcal{O}(\sqrt{n}) \subset \mathcal{O}(n)$?	✓
$2^n \notin \mathcal{O}(\exp(n))$?	✗

Time Consumption

Assumption 1 Operation = $1\mu s$.

problem size	1	100	10000	10^6	10^9
$\log_2 n$	$1\mu s$				
n	$1\mu s$				
$n \log_2 n$	$1\mu s$				
n^2	$1\mu s$				
2^n	$1\mu s$				

Time Consumption

Assumption 1 Operation = $1\mu s$.

problem size	1	100	10000	10^6	10^9
$\log_2 n$	$1\mu s$	$7\mu s$	$13\mu s$	$20\mu s$	$30\mu s$
n	$1\mu s$				
$n \log_2 n$	$1\mu s$				
n^2	$1\mu s$				
2^n	$1\mu s$				

Time Consumption

Assumption 1 Operation = $1\mu s$.

problem size	1	100	10000	10^6	10^9
$\log_2 n$	$1\mu s$	$7\mu s$	$13\mu s$	$20\mu s$	$30\mu s$
n	$1\mu s$	$100\mu s$	$1/100s$	$1s$	17 minutes
$n \log_2 n$	$1\mu s$				
n^2	$1\mu s$				
2^n	$1\mu s$				

Time Consumption

Assumption 1 Operation = $1\mu s$.

problem size	1	100	10000	10^6	10^9
$\log_2 n$	$1\mu s$	$7\mu s$	$13\mu s$	$20\mu s$	$30\mu s$
n	$1\mu s$	$100\mu s$	$1/100s$	$1s$	17 minutes
$n\log_2 n$	$1\mu s$	$700\mu s$	$13/100\mu s$	$20s$	8.5 hours
n^2	$1\mu s$				
2^n	$1\mu s$				

Time Consumption

Assumption 1 Operation = $1\mu s$.

problem size	1	100	10000	10^6	10^9
$\log_2 n$	$1\mu s$	$7\mu s$	$13\mu s$	$20\mu s$	$30\mu s$
n	$1\mu s$	$100\mu s$	$1/100s$	$1s$	17 minutes
$n \log_2 n$	$1\mu s$	$700\mu s$	$13/100\mu s$	$20s$	8.5 hours
n^2	$1\mu s$	$1/100s$	1.7 minutes	11.5 days	317 centuries
2^n	$1\mu s$				

Time Consumption

Assumption 1 Operation = $1\mu s$.

problem size	1	100	10000	10^6	10^9
$\log_2 n$	$1\mu s$	$7\mu s$	$13\mu s$	$20\mu s$	$30\mu s$
n	$1\mu s$	$100\mu s$	$1/100s$	$1s$	17 minutes
$n \log_2 n$	$1\mu s$	$700\mu s$	$13/100\mu s$	$20s$	8.5 hours
n^2	$1\mu s$	$1/100s$	1.7 minutes	11.5 days	317 centuries
2^n	$1\mu s$	10^{14} centuries	$\approx \infty$	$\approx \infty$	$\approx \infty$

A good strategy?

... Then I simply buy a new machine

A good strategy?

... Then I simply buy a new machine If today I can solve a problem of size n, then with a 10 or 100 times faster machine I can solve ...

Komplexität	(speed $\times 10$)	(speed $\times 100$)
$\log_2 n$		
n		
n^2		
2^n		

A good strategy?

... Then I simply buy a new machine If today I can solve a problem of size n, then with a 10 or 100 times faster machine I can solve ...

Komplexität	(speed $\times 10$)	(speed $\times 100$)
$\log_2 n$	$n \to n^{10}$	$n \to n^{100}$
n		
n^2		
2^n		

A good strategy?

... Then I simply buy a new machine If today I can solve a problem of size n, then with a 10 or 100 times faster machine I can solve ...

Komplexität	(speed $\times 10$)	(speed $\times 100$)
$\log_2 n$	$n \rightarrow n^{10}$	$n \rightarrow n^{100}$
n	$n \rightarrow 10 \cdot n$	$n \rightarrow 100 \cdot n$
n^2		
2^n		

A good strategy?

... Then I simply buy a new machine If today I can solve a problem of size n, then with a 10 or 100 times faster machine I can solve ...

Komplexität	(speed $\times 10$)	(speed $\times 100$)
$\log_2 n$	$n \to n^{10}$	$n \to n^{100}$
n	$n \to 10 \cdot n$	$n \to 100 \cdot n$
n^2	$n \to 3.16 \cdot n$	$n \to 10 \cdot n$
2^n		

A good strategy?

... Then I simply buy a new machine If today I can solve a problem of size n, then with a 10 or 100 times faster machine I can solve ...

Komplexität	(speed $\times 10$)	(speed $\times 100$)
$\log_2 n$	$n \to n^{10}$	$n \to n^{100}$
n	$n \to 10 \cdot n$	$n \to 100 \cdot n$
n^2	$n \to 3.16 \cdot n$	$n \to 10 \cdot n$
2^n	$n \to n + 3.32$	$n \to n + 6.64$

Asymptotic Running Times with Θ

```
void run(int n){
  for (int i = 1; i<n; ++i)
    for (int j = 1; j<n; ++j)
      op();
}
```

How often is `op()` called?

Asymptotic Running Times with mit Θ

```
void run(int n){
  for (int i = 1; i<n; ++i)
    for (int j = i; j<n; ++j)
      op();
}
```

How often is `op()` called?

Asymptotic Running Times with Θ

```
void run(int n){
  for (int i = 1; i<n; ++i){
    op();
    for (int j = i; j<n; ++j)
      op();
    }
}
```

How often is `op()` called?

Asymptotic Running Times with Θ

```
void run(int n){
  for(int i = 1; i <= n; ++i)
    for(int j = 1; j*j <= n; ++j)
      for(int k = n; k >= 2; --k)
        op();
}
```

How often is `op()` called?

3. Programming Exercise

Unfair Dice

Dice Simulation

Given: Simulation of the uniformly distributed random variable using `Math.Random()`

We want: Simulation of a *fair* die

Dice Simulation

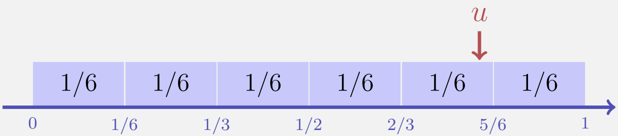
`Math.Random()` returns $U \in [0, 1)$ with

$$\mathbb{P}(U \in [l, r)) = r - l.$$

`Dice()` should return $Y \in \{1, \ldots, 6\}$, such that

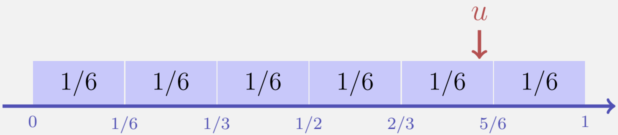
$$\mathbb{P}(Y = k) = 1/6 \text{ for all } k \in \{1, \ldots, 6\}$$

Cumbersome, but Correct




```
static int Dice(){
  double u = Math.random();
  if (u<1.0/6) return 1;
  else if (u<1.0/3) return 2;
  else if (u<1.0/2) return 3;
  else if (u<2.0/3) return 4;
  else if (u<5.0/6) return 5;
  else return 6;
}
```

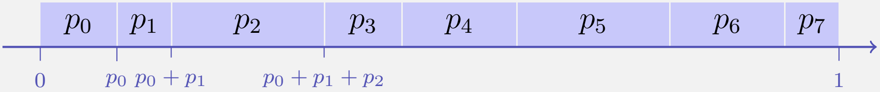
Easier




```
static int Dice(){
  double u = Math.random();
  return (int)(u*6+1);
}
```

Toss an Unfair Dice

Given: Probabilitiy vector $p = (p_0, \ldots, p_{n-1})$ with $\sum_{i=0}^{n-1} p_i = 1$ and $p_i \geq 0$ $(0 \leq i < n)$.

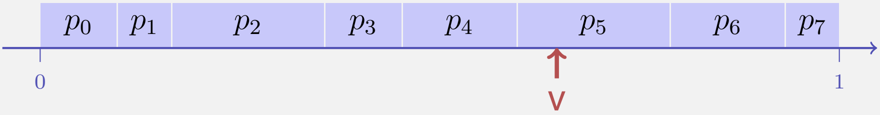


Wanted: `Sample(p)` returning j $(0 \leq j < n)$ with probability p_j.

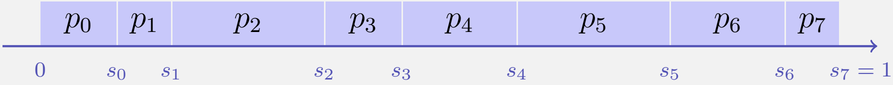
Formal

Using an existing random number generator, a number v is drawn uniformly distributed from the interval $[0, 1)$. From this number v an integer $0 \leq S(p, v) < n$ is generated according to the following rule

$$S(p, v) = \min\{0 \leq i < n : \sum_{k=0}^{i} p_k > v\}$$



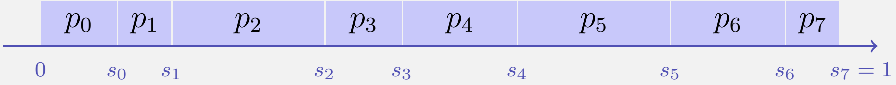
Toss an Unfair Dice




```
static int Sample(double[] p){
  double u = Math.random();
  if (u<p[0]) return 0;
  if (u<p[0]+p[1]) return 1;
  if (u<p[0]+p[1]+p[2]) return 2;
  if (u<p[0]+p[1]+p[2]+p[3]) return 3;
  ...
}
```

Zu umständlich: wir brauchen eine Schleife!

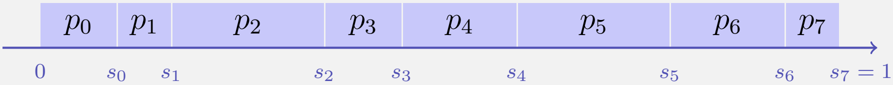
A Tiny Trick




```
static int Sample(double[] p){
  double u = Math.random();
  if (u<p[0]) return 0;
  u -= p[0];
  if (u<p[1]) return 1;
  u -= p[1];
  if (u<p[2]) return 2;
  ...
}
```

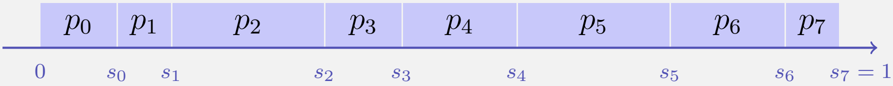
We do not have to compute the sums of the p_i

In a Loop




```
static int Sample(double[] p){
  double u = Math.random();
  for (int k = 0; k < p.length-1; ++k){
    if (u<p[k]) return k;
    u -= p[k];
  }
  return p.length-1;
}
```

More Compact




```
static int Sample(double[] p){
  double u = Math.random();
  int k=0;
  while (k < p.length && u>0){
    u -= p[k++];
  }
  return k-1;
}
```

Questions or Comments?