

Randomisierte Algorithmen – Übungen

1. Von Monte Carlo nach Las Vegas

Gegeben sei ein Monte-Carlo-Algorithmus M für ein Problem Π , welcher für jede Instanz des Problems durchschnittlich T Schritte benötigt und dabei eine korrekte Antwort mit Wahrscheinlichkeit α liefert. Angenommen für Π existiere ein deterministischer Verifikationsalgorithmus V , der in K Schritten feststellt, ob eine mögliche Lösung stimmt. Transformieren Sie M in einen Las-Vegas-Algorithmus L , welcher immer eine richtige Lösung für Π liefert, und zeigen Sie, dass die erwartete Laufzeit $(T + K)/\alpha$ Schritte ist.

2. Zufallserzeugung

Gegeben sei eine Methode $RBIT()$, welche bei jedem Aufruf ein zufälliges, uniform verteiltes Bit $R \in \{0, 1\}$ liefert. Konstruieren Sie daraus eine Methode (in Pseudocode), welche Zufallswerte mit einer anderen Verteilung erzeugt, sofern die Methode terminiert. Es darf sein, dass Ihre Lösung mit kleiner Wahrscheinlichkeit (WSK) nicht terminiert.

- a) Beschreiben Sie eine Methode $Alpha()$, welche ein Bit $A \in \{0, 1\}$ erzeugt, so dass $A = 1$ mit WSK $\frac{5}{8}$ und $A = 0$ sonst.
- b) Beschreiben Sie eine Methode $Beta()$, welche $B \in \{1, 2, 3\}$ erzeugt, so dass $B = 1$ mit WSK $\frac{1}{13}$, $B = 2$ mit WSK $\frac{4}{13}$ und $B = 3$ mit WSK $\frac{8}{13}$.

1.) $M(x) \rightarrow ?$ mit α ist $M(x) \rightarrow y$ in T -Schritten
 $V(x) \rightarrow y/n$ in K -Schritten
 $M(x)$ wird so lange wiederholt bis $M(x) \rightarrow y \Rightarrow V(y) \Rightarrow \text{return } y$ $M(x)$ wiederholen bis $V(M(x)) = 1$ ausgibt
 M benötigt T -Schritte um $M(x)$ zu berechnen
 V braucht K -Schritte um $y = M(x)$ zu verifizieren
 1 Durchlauf benötigt $T + K$ Schritte allerdings stimmt das Ergebnis nur mit Wahrscheinlichkeit α Dies wird wiederholt bis das Ergebnis stimmt \Rightarrow Erwartungswert ist anz. Wied. bis Erfolg : $1/\alpha$
 $\Rightarrow \frac{T + K}{\alpha}$

2.) a)
 $A = 1 \quad \frac{5}{8}$
 $A = 0 \quad \frac{3}{8}$
 $RBIT$ liefert 0 oder 1 mit Wahren 50%

0	1	2	3
---	---	---	---

 $alpha() : WSK \frac{5}{8}$ für 1, $WSK \frac{3}{8}$ für 0

```
Int counter = 0
For (int a = 0; a < 8; a++)
{counter += RBIT();}
If (counter >= 5)
{Return 1;}
Else
Return 0;
```

b) $Beta() : WSK \frac{1}{13}$ für 1, $WSK \frac{4}{13}$ für 2, $WSK \frac{8}{13}$ für 3

```
Int counter = 0
For (int a = 0; a < 13; a++)
{counter += RBIT();}
If (counter >= 8)
{Return 3;}
Else if (counter >= 4)
Return 2;
Else
Return 1;
```