Liste der Probleme und Algorithmen

AA	Approximationsalgorithmus
MCA	Monte-Carlo-Algorithmus (mit... 1F: einseitigem Fehler, 2F: beidseitigem Fehler, UF: unbeschränktem Fehler)

Probleme	*Algorithmen*
Approximative Algorithmen	
MinVC *Eingabe*: G = (V, E)	• **Einfacher 2-AA** $O(\lvert E\rvert)$
MinSetCover (NP-schwer) *Eingabe*: Menge X, Mengensystem $\mathscr{F}$	• **GreedySetCover** $O(\lvert X\rvert^2 \cdot \lvert\mathscr{F}\rvert)$, H(m)-AA mit $m = \max\{\lvert F\rvert : F \in \mathscr{F}\}$
Einfaches Rucksackproblem (**SKP**, entspricht KP_0]) *Eingabe*: n Gewichte	• **Greedy-SKP** 2-AA, $O(n \cdot \log n)$ • **P-SKP** PTAS, $O(n^{\lceil 1/\varepsilon\rceil+1} \cdot \log n)$
Eingeschränktes Rucksackproblem (**KP_δ**)	• **MOD-SKP** PTAS für festes δ, Laufzeit exponentiell in $\lceil 1/\varepsilon\rceil$
Allgemeines Rucksackproblem (**KP**) *Eingabe*: n Gewichte mit ihren Kosten c_i Es gilt: KP $\in$ NPO(I)	• **DPKP** optimal, $O(n^2 \cdot \max\{c_1, ..., c_n\})$ (pseudopolynomiell) • **F-KP** FPTAS, $O(1/\varepsilon \cdot n^3)$
Δ-TSP *Eingabe*: G = ((V, E), c)	• **MIN-ST mit Tiefensuche** 2-AA, $O(\lvert E\rvert \cdot \log(\lvert E\rvert))$ • **Christofides** 1,5-AA, $O(\lvert E\rvert^4)$
Randomisierte Algorithmen	
Sortieren *Eingabe*: n Zahlen	• **Random QuickSort, RQS** $E[T] \sim 2n \cdot \ln(n) + \Theta(n)$ Las Vegas
MaxE*k*Sat *Eingabe*: Formel $F = F_1 \wedge ... \wedge F_m$ über n Variablen	• **Random Assignment** $E[R(F)] \leq 2^k / (2^k - 1)$
MaxCut *Eingabe*: G = (V, E)	• **RanSam** $E[R(G)] \leq 2$
Quadratische Nichtreste *Eingabe*: Primzahl p	• **Quadratic Nonresidue** Las Vegas

Probleme	*Algorithmen*
(Nicht-)Primzahl-Test	*Gemeinsames Prinzip*: **Methode häufiger Zeugen** • **SSSA (vereinfachter Solovay-Strassen-Alg.)** *Eingabe*: ungerade Zahl n mit ungeradem (n-1)/2 *Zeugenkriterium*: $a^{(n-1)/2} \notin \{-1;1\}$ MCA mit 1F, polynomielle Laufzeit • **SSA (Solovay-Strassen-Algorithmus)** *Eingabe*: ungerade Zahl n *Zeugenkriterium*: $a^{(n-1)/2} \neq \mathrm{Jac}[{}^a/_n]$ MCA mit 1F, polynomielle Laufzeit • **Miller-Rabin** *Eingabe*: ungerade Zahl n *Zeugenkriterium*: Wurzelzeuge MCA mit 1F, polynomielle Laufzeit
Primzahlerzeugung	• **Prime Generation PG(l, k)** *Eingabe*: gewünschte Länge l, Anzahl k von Iterationen eines Primzahltests W'keit. Suche erfolglos: $< e^{-l}$ W'keit. falscher Ausgabe: $< l^2 / (2^{l-1})$
Nicht-Äquivalenz von Polynomen *Eingabe*: Polynome über $\mathbb{Z}_n$	• **NEQ-POL** *Prinzip*: Fingerprinting
MinCut *Eingabe*: Multigraph G = (V, E)	• **Random Contraction (RC)** $O(n^2)$, Prob[opt. Schnitt] $\geq 2 / (n \cdot (n-1)) = 1/$[n über 2] *Erweiterung*: **RC_k** für k Läufe von RC (bei $k = \frac{1}{2} n^2$ ist Prob[opt. Schnitt] $> 1 - 1/e$) • **l-COMB-CONTRACT** mit l: $\mathbb{N} \rightarrow \mathbb{N}$ (Übergang zur det. Berechnung) $O(n^2 + l(n)^3)$ Erfolgsw'keit: [l(n) über 2] / [n über 2] • **RRC** $O(n^2 \cdot \log(n))$ Erfolgsw'keit $\geq 1 / \Omega(\log_2(n))$
MaxSat	• **RSMS** (Random-Sampling für MaxSat) • **RRRMS** (Relaxation mit Random-Rounding für MaxSat) polynomielle Laufzeit $E[R_{RRRMS}(F)] \leq e / (e-1)$ $E[R_{RRRMS}(F)] \leq k^k / (k^k - (k-1)^k)$ (MaxEkSat) • **COMB** $E[R_{COMB}(F)] \leq 4/3$ • **RED(RSMS)**
3Sat	• **Schöning** MCA mit 1F $O(\lvert F \rvert \cdot n^{3/2} \cdot (4/3)^n)$ Fehlerwahrscheinlichkeit $< e^{-10}$