S9

Exercise S9.1 – *Boosting the success probability of Monte Carlo Algorithms*

Alice, Bob and Claire have learned that Monte Carlo algorithms can often be improved by running them multiple times. They are each given a blackbox [1] Monte Carlo algorithm $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ with a certain bound on the error. The task is to construct new algorithms $\mathcal{A}\_improved$, $\mathcal{B}\_improved$, and $\mathcal{C}\_improved$ resp. (using $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ resp. as subroutine) with a success probability of at least 99%. Try to keep the number of calls to the subroutines small.

(a) Given a graph (V, E), Alice tries to find a vertex set $\emptyset \neq S \subsetneq V$ that minimizes $|\delta(S)|$. She is given a Monte Carlo Algorithm $\mathcal{A}$ that, given a graph, returns some vertex set $\emptyset \neq S \subsetneq V$. With probability at least $p_{\mathcal{A}} \geq 1/\binom{n}{2}$ this set minimizes $|\delta(S)|$. Use this algorithm $\mathcal{A}$, to construct another algorithm $\mathcal{A}\_improved$ for the same problem, that has a success probability of at least 99%. *Hint:* $1 + x \leq e^x$ *for* $x \in \mathbb{R}$.

$\varepsilon = 1/\binom{n}{2} \approx 2/n(n-1)$

One-sided MC rep. (on CS), Satz 2.76 (minimisation problem):
For $N = \varepsilon^{-1} \ln \delta^{-1}$, N-fold repetition boost success prob to $\geq 1-\delta$

$\delta = 0.01 \quad 0.99 = 1 - 0.01$

$N := \lceil (2/n(n-1))^{-1} \ln((0.01)^{-1}) = \frac{n(n-1)}{2} \ln 100 \rceil$

$\mathcal{A}\_improved$: Calls $\mathcal{A}$ N-times and return smallest (best) result

Satz 2.76. Sei $\varepsilon > 0$ und $\mathcal{A}$ ein randomisierter Algorithmus für ein Maximierungsproblem, wobei gelte:

$$\Pr[\mathcal{A}(I) \geq f(I)] \geq \varepsilon.$$

$\leq f(I) = \operatorname{argmin}_{S \subsetneq V} |\delta(S)|$

Dann gilt für alle $\delta > 0$: bezeichnet man mit $\mathcal{A}_\delta$ den Algorithmus, der $N = \varepsilon^{-1} \ln \delta^{-1}$ unabhängige Aufrufe von $\mathcal{A}$ macht und die *beste* der erhaltenen Antworten ausgibt, so gilt für den Algorithmus $\mathcal{A}_\delta$, dass

$$\Pr[\mathcal{A}_\delta(I) \geq f(I)] \geq 1 - \delta.$$

(Für Minimierungsprobleme gilt eine analoge Aussage wenn wir „$\geq f(I)$“ durch „$\leq f(I)$“ ersetzen.)

(b) Bob wants to check if a number is prime. He already knows a Monte Carlo algorithm $\mathcal{B}$ which takes as input a natural number N. If N is prime, the algorithm always returns 'prime'. If N is not prime, the algorithm returns 'not a prime' with probability $p_B \geq 1/2$. Use Bobs algorithm $\mathcal{B}$, to construct another algorithm $\mathcal{A}\_improved$ for the same problem, that has a success probability of at least 99%. $= 1-\delta$

$\varepsilon = 1/2$, $\delta = 0.01$

One-sided MC rep. (on CS), Satz 2.74:
For $N = \varepsilon^{-1} \ln \delta^{-1}$, N-fold repetition boosts success prob to $\geq 1-\delta$

$N := \lceil (1/2)^{-1} \ln(0.01)^{-1} \rceil = \lceil 2 \ln 100 \rceil$

$\mathcal{B}$-improved: Repeat $\mathcal{B}$ N-times and return "not prime" as soon as "not prime" was returned once, otherwise return "prime"

(c) Claire chose the most difficult problem. She wants to determine if a given graph has an Hamiltonian cycle. She thought of an algorithm $\mathcal{C}$ that, given a graph, outputs 'YES' or 'NO'. If a graph G has (does not have) a Hamiltonian cycle, the algorithms $\mathcal{C}(G)$ returns 'YES' ('No') with probability $p_C \geq 3/4$. Help Claire to find an algorithm C_*improved* that is correct with probability at least 99%.

$\varepsilon = 1/4$, $p_C \geq 3/4 = 1/2 + \varepsilon$, $\delta = 0.01$

"Regular" MC Repetition on CS, Satz 2.75.:
For $N = 4\varepsilon^{-2} \ln \delta^{-1}$ N-fold rep. boosts success prob. from $\geq 1/2+\varepsilon$ to $\geq 1-\delta$

$N = \lceil 4 (1/4)^{-2} \ln(0.01)^{-1} \rceil = 4 \cdot 16 \cdot \ln 100 = \lceil 64 \ln 100 \rceil$

C-improved: Call C N-Times and return the value (Y/N) that was returned by C more often