

Aufgabe 1 – *Broken servers*

Sie sind mit einem Netzwerk verbunden, das aus n Servern besteht, die von 1 bis n nummeriert sind. Sie können jeden Server i in Zeit $\mathcal{O}(1)$ kontaktieren und erhalten als Antwort entweder eine ‘0’ oder eine ‘1’. Leider sind einige der Server kaputt und Sie sollen herausfinden welche Server betroffen sind.

- Falls Server i kaputt ist, sendet er bei jeder Anfrage ein unabhängig gleichverteiltes Bit.
 - Falls Server i intakt ist, dann antwortet er auf jede Anfrage mit dem gleichen Bit $a_i \in \{0, 1\}$. Allerdings ist der Wert a_i unbekannt und kann von Server zu Server variieren.
- (a) Seien $\delta > 0$ und $i \in [n]$ gegeben. Beschreiben Sie einen Monte-Carlo Algorithmus, der herausfindet, ob Server i kaputt ist. Berechnen Sie die Fehlerwahrscheinlichkeiten (abhängig davon ob der Server kaputt/intakt ist) Ihres Algorithmus und stellen Sie sicher, dass Ihr Algorithmus Fehlerwahrscheinlichkeit höchstens $\frac{\delta}{n}$ hat.
- (b) Falls ein Server i kaputt (bzw. intakt) ist, ist es dann sicher, dass Ihr Algorithmus aus (a) dies herausfindet? Umgekehrt, wenn Ihr Algorithmus ausgibt, dass ein Server i kaputt (bzw. intakt) ist, können Sie sich sicher sein, dass diese Ausgabe korrekt ist?
- (c) Sei $\delta > 0$ gegeben. Beschreiben Sie einen Monte-Carlo Algorithmus, der eine Liste aller kaputten Server erstellt und Fehlerwahrscheinlichkeit höchstens δ hat. Hierbei sagen wir, dass der Algorithmus erfolgreich ist, wenn die Liste alle kaputten Server und keinen intakten Server enthält.