A&W Lernziele

DISCLAIMER: Dieses Dokument dient als ergänzende Lernhilfe und ist keine offizielle Kursveröffentlichung. Es wurde von Josia Heger (TA) erstellt und gibt seine persönlichen Empfehlungen zu Lernstrategien und wichtigen Konzepten wieder. Es wird nicht garantiert, dass die Prüfung nur die hier aufgeführten Themen abdeckt.

Allgemeine Ratschläge:

- Graphen-Algorithmen (Hopcroft-Karp, Euler-Tour, Hamiltonkreis, ...) muss man nicht programmieren können. Es reicht, wenn man den Pseudocode grob wiedergeben kann. Probiert die Algorithmen trotzdem so gut zu verstehen, dass ihr sie jemand anderem erklären könntet.
- Falls mehrere Zufallsvariablen vorkommen, sollte man sich immer als erstes überlegen, ob sie abhängig oder unabhängig sind, und dann entscheiden, welche Sätze überhaupt anwendbar sind.
- Ich empfehle, das gesamte Skript und die Mini-Quizzes bei Gemini (oder einem anderen AI Model) hochzuladen, um weitere Aufgaben zu generieren im selben Stil der Mini-Quizzes.
- Für den Satz von Bayes und den Satz der Totalen Wahrscheinlichkeit ist es sehr nützlich, die Formel insbesondere für den Spezialfall, wenn man $\Omega = A \cup \bar{A}$ nur auf zwei Ereignisse aufteilt, auswendig zu können.
- Löst alle alten Code Expert Aufgaben (auch mehrmals, falls nötig), und verwendet dabei KEINE AI. Ihr müsst an der Prüfung alles selber debuggen. Es bringt nichts, wenn ihr 95% selber programmieren könnt und dann die restlichen "kleinen" Fehler mit AI löst. Ihr müsst die Aufgaben vollständig selber lösen können.

Mathematische Grundlagen

☐ Definitionen kennen von: disjunkt, kartesisches Produkt, Binomialkoeffizient, $\mathbb{N} = \{1, 2, 3, \dots\}$, $[n] = \{1, 2, \dots, n\}$

☐ Potenz- und Logarithmusgesetze kennen.

☐ Abschätzung $e^x \geq 1 + x$ kennen.

☐ Verschiedene Beweismuster aus DiskMat kennen (direkt, Widerspruchsbeweis, Kontraposition, Induktion, ...)

☐ Mit der O-Notation rechnen können.

Graphentheorie Grundlagen

☐ Definitionen kennen von: Graph, Knoten, Kanten, Grad, Weg, Pfad, Kreis, Zyklus, Länge, adjazent, inzident, Nachbarschaft, k-regulär, Handschlag-Lemma für gerichtete und für ungerichtete Graphen, induzierter Teilgraph, zusammenhängend, Zusammenhangskomponenten, Baum, Wald, vollständig, MST, Adjazenzmatrix, Adjazenzliste

☐ DFS/BFS gut verstehen.

Zusammenhang

☐ Definitionen kennen von: Knotenseparator, Kantenseparator, k-(Knoten)-zusammenhängend, k-Kanten-zusammenhängend.

☐ Wissen dass Knotenzusammenhang $\leq$ Kantenzusammenhang $\leq$ minimaler Grad.

☐ Satz von Menger kennen und in beide Richtungen anwenden können.

☐ Definitionen von Artikulationsknoten und Brücken kennen.

☐ Wissen wie der Tarjan Algorithmus funktioniert (der die Artikulationsknoten und Brücken findet), die Laufzeit, und die Definitionen für dfs- und low-Zahlen, Baumkanten und Restkanten.

☐ Wissen, welche Bedingungen für die dfs- und low Zahlen genau dann gelten, wenn ein Knoten ein Artikulationsknoten ist, oder eine Kante eine Brücke ist.

☐ Definition der Block-Äquivalenzrelation kennen, und für einen Graphen die Block-Zerlegung zeichnen können und deren Eigenschaften (kreisfrei, bipartit, zusammenhängend wenn es der originale Graph war) kennen.

Kreise

☐ Definition der Eulertour kennen. Wissen dass es eine Eulertour gibt, genau dann wenn alle Grade gerade sind und alle Kanten in derselben Zusammenhangskomponente liegen.

☐ Algorithmus zum finden der Eulertour grob erklären können und die Laufzeit kennen.

☐ Definition des Hamiltonkreises kennen.

☐ Spezialfälle kennen, für die wir direkt sagen können, ob es einen Hamiltonkreis gibt: $m \times n$-Gittergraphen, d-dimensionale Hyperwürfel, Bipartite Graphen $G = A \uplus B$ mit $|A| \neq |B|$

☐ Exponentiellen DP Algorithmus zum finden eines Hamiltonkreises kennen, d.h. Teilproblem, Base Case, Rekursion, Auslesen der Lösung und Laufzeit.

☐ Satz von Dirac kennen, inklusive Beweis und Laufzeit.

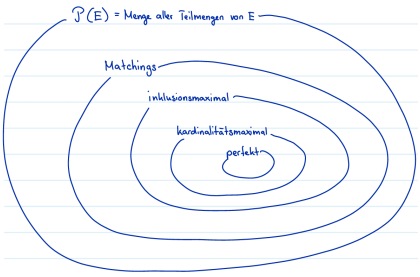
☐ Definition des Travelling Salesman Problem kennen und wissen, dass es NP-vollständig ist und es auch keinen Approximationsalgorithmus dafür gibt.

☐ Definition des metrischen TSP kennen, und den 2- und $\frac{3}{2}$- Approximationsalgorithmus schrittweise erklären können und die Beweise und Laufzeiten dafür kennen.

Matchings

☐ Definition eines Matchings kennen.

☐ Die Definitionen der drei Eigenschaften inklusionsmaximal, kardinalitäs-maximal und perfekt kennen und die “Hierarchie”:



☐ Greedy Matching-Algorithmus erklären können, die Laufzeit kennen, und wissen dass $|M_{\text{Greedy}}| \geq \frac{|M_{\max}|}{2}$

☐ Definition eines augmentierenden Pfades kennen, und wissen wie man ein Matching mit einem augmentierenden Pfad vergrössern kann.

☐ Satz von Berge kennen, inklusive Beweis.

☐ Hopcroft-Karp Algorithmus Schritt für Schritt genau erklären können. (dafür könnte die Visualisierung auf josia-heger.com/algorithms hilfreich sein) Man sollte auch den Pseudocode (vom Skript) dafür können, die Korrektheit erklären können, und den Beweis der Laufzeit grob verstehen.

☐ Satz von Hall kennen und anwenden können. Den Beweis sollte man grob verstehen.

☐ Wissen dass man k-reguläre Graphen in perfekte Matchings zerlegen kann, und die Laufzeit kennen, in der ein perfektes Matching bestimmt werden kann.

Färbungen

- [ ] Definition einer Färbung und der chromatischen Zahl kennen.
- [ ] Chromatische Zahl insbesondere für Kreise, vollständige Graphen und Bäume kennen.
- [ ] Wissen, dass das Problem für $k \geq 3$ NP-vollständig ist, sowie auch jeder Approximationsalgorithmus
- [ ] Definition von bipartit und k-partit kennen, und deren Zusammenhang mit einer 2-Färbung oder einer k-Färbung.
- [ ] Wissen, dass ein Graph genau dann bipartit ist, wenn es keinen Kreis ungerader Länge gibt.
- [ ] Definition von independent Sets kennen und wissen, dass man Farbklassen tauschen kann.
- [ ] Greedy-Färbung Algorithmus kennen und wissen, dass höchstens $\Delta(G) + 1$ viele Farben gebraucht werden, und es eine optimale Reihenfolge gibt, die genau $\chi(G)$ viele Farben verwendet. Man sollte auch die Laufzeit kennen.
- [ ] Heuristik zur Bestimmung der Reihenfolge kennen (aka. Smallest Last Coloring), und wissen, dass falls es in jedem induzierten Subgraphen einen Knoten mit Grad $\leq k$ gibt, höchstens $k + 1$ Farben benötigt werden.
- [ ] Definition von planaren Graphen kennen und wissen, dass jeder planare Graph einen Knoten mit Grad ≤ 5 hat, und jeder planare Graph 4-färbbar ist.
- [ ] Den Algorithmus zur Färbung von 3-färbbaren Graphen mit $\mathcal{O}(\sqrt{|V|})$ Farben erklären können.
- [ ] Satz von Brooks kennen und anwenden können.
- [ ] Wissen dass die chromatische Zahl unabhängig ist von der Länge eines kürzesten Kreises.

Wahrscheinlichkeit

- [ ] Definitionen kennen von: Wahrscheinlichkeitsraum, Elementarereignis, Ereignis, Komplementärereignis, Wahrscheinlichkeit eines Ereignisses.
- [ ] Grundlegende Eigenschaften kennen, wie z.B. $\Pr[\varnothing] = 0$, $\Pr[\Omega] = 1$, $\Pr[\overline{A}] = 1 - \Pr[A]$, $A \subseteq B \implies \Pr[A] \leq \Pr[B]$
- [ ] Definition von Laplace-Raum kennen, und bestimmen können, ob etwas ein Laplace-Raum ist oder nicht.

Für die Folgenden Sätze sollte man sicherlich ein intuitives Verständnis haben, und die Beweise kennen (ausser für die Siebformel).

  - [ ] Additionssatz
  - [ ] Siebformel (ist auf dem Cheatsheet)
  - [ ] union bound (ist auf dem Cheatsheet)
  - [ ] Multiplikationssatz (ist auf dem Cheatsheet)
  - [ ] Satz der Totalen Wahrscheinlichkeit (ist auf dem Cheatsheet)
  - [ ] Satz von Bayes (ist auf dem Cheatsheet)

- [ ] Definition für bedingte Wahrscheinlichkeit kennen (ist auf dem Cheatsheet)
- [ ] Definition der Unabhängigkeit von Ereignissen kennen.
- [ ] Allgemeinere Definition der Unabhängigkeit von Ereignissen kennen (Lemma 2.23).
- [ ] Wissen, was für die Vereinigung und die Schnittmenge von unabhängigen Ereignissen gilt (Lemma 2.24).
- [ ] Eigenschaften wie z.B. $\Pr[A \mid B] \cdot \Pr[B] = \Pr[B \mid A] \cdot \Pr[A]$ oder $\Pr[A \mid \bar{A}] = 0$ durch einsetzen der Definitionen beweisen können.
- [ ] Formeln der Kombinatorik kennen:

Kombinatorik: wir ziehen k Elemente aus einer Menge mit n Elementen

	geordnet (Reihenfolge wichtig)	ungeordnet (Reihenfolge egal)
mit zurücklegen (mit Wiederholungen)	n^k	$\binom{n+k-1}{k}$
ohne zurücklegen (ohne Wiederholungen)	$n^{\underline{k}} = \frac{n!}{(n-k)!} = n \cdot (n-1) \cdot \ldots \cdot (n-k+1)$	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$

- [ ] Definition einer Zufallsvariable kennen.
- [ ] Notation für Ereignisse wie z.b: $X \leq x \overset{\text{def}}{=} \{\omega \in \Omega \mid X(\omega) \leq x\}$ kennen.
- [ ] Definition einer Indikatorvariablen und deren Erwartungswert kennen.
- [ ] Definition der Dichtefunktion und der Verteilungsfunktion kennen, und diese für eine gegebene Zufallsvariable berechnen können.
- [ ] Drei Definitionen des Erwartungswertes kennen (Def. 2.27, Lemma 2.29, Satz 2.30)
- [ ] Linearität des Erwartungswertes kennen und anwenden. (ist auf dem Cheatsheet)
- [ ] Definition einer bedingten Zufallsvariable kennen.
- [ ] Wissen, wie man den Erwartungswert auf mehrere Ereignisse aufteilen kann (Satz 2.32). Dieser Satz wurde in der Vorlesung nicht sehr stark gewichtet, aber er ist sehr praktisch für Code Expert.
- [ ] Zwei Definitionen der Varianz (Def. 2.39 und Satz 2.40) und Standardabweichung kennen. (ist auf dem Cheatsheet)
- [ ] Wissen dass $\mathrm{Var}[a \cdot X + b] = a^2 \cdot \mathrm{Var}[X]$. Solche Sachen sollte man auch intuitiv gut verstehen. (ist auf dem Cheatsheet)

Für die folgenden Verteilungen sollte man jeweils die Definition, den Erwartungswert, die Varianz, gegebenenfalls die Formel für die Dichte- und Verteilungsfunktion, und weitere Eigenschaften kennen. Die genauen Details, was man können sollte findet sich hier auf Seite 7-11: https://www.josia-heger.com/lessons/AuW_FS26/week-7.pdf. Viele Formeln dazu sind auch auf den Cheatsheet.

- [ ] Bernoulli-Verteilung
- [ ] Binomial-Verteilung
- [ ] Poisson-Verteilung
- [ ] Geometrische Verteilung
- [ ] Negativ Binomial Verteilung

- [ ] Definition der gemeinsamen Verteilung, der Randverteilung und deren Zusammenhang kennen und berechnen können.
- [ ] Zwei Definitionen von Unabhängigkeit für Zufallsvariablen kennen (Def. 2.52, Lemma 2.53). (ist auf dem Cheatsheet)
- [ ] Wissen, dass Unabhängigkeit von Zufallsvariablen durch das Anwenden von Funktionen erhalten bleibt (Satz 2.55). (ist auf dem Cheatsheet)
- [ ] Wissen, dass die Multiplizität des Erwartungswertes und die Additivität der Varianz im allgemeinen nur für unabhängige Zufallsvariablen gilt. (ist auf dem Cheatsheet)
- [ ] Faltungsformel für zwei unabhängige Zufallsvariablen kennen (Satz 2.58). (ist auf dem Cheatsheet)
- [ ] Waldsche Identität kennen. (ist auf dem Cheatsheet)
- [ ] Markov-/Chebyshev-/Chernoff-Ungleichungen kennen und anwenden können. (ist auf dem Cheatsheet)

Am Anfang fühlt sich das Anwenden von Chebyshev und Chernoff schwierig an. Jedoch folgen die Umformungsschritte immer exakt demselben Muster. Wenn man dieses Muster 3-4 Mal angewendet hat, wird es viel einfacher.

☐ Obwohl die folgenden Aufgaben wahrscheinlich nicht direkt abgefragt werden, ist es trotzdem ratsam, die ungefähre Herangehensweise zu kennen: Ziegenproblem, Coupon Collector, Geburtstagsproblem aka. Balls into Bins Problem, Stabile Mengen

Randomisierte Algorithmen

☐ Definitionen von Las-Vegas und Monte-Carlo Algorithmen kennen, und entscheiden, zu welcher Kategorie ein gegebener Algorithmus gehört.

☐ Wissen wie ein Las-Vegas Algorithmus zu einem Monte-Carlo Algorithmus umgewandelt werden kann.

☐ Target-Shooting Algorithmus kennen, und den Beweis grob kennen. (ist auf dem Cheatsheet)

☐ Wissen in welchen Fällen man Fehlerreduktion verwenden kann, und die entsprechende Anzahl Wiederholungen bestimmen. (ist teilweise auf dem Cheatsheet)

☐ Laufzeit von Randomisiertem QuickSort und QuickSelect kennen und die Algorithmen erklären können.

☐ Grundlagen von Modularer Arithmetik aus DiskMat ungefähr kennen, z.B. ggT, $\mathbb{Z}_n^*$, Körper, kleiner Fermat'scher Satz, Satz von Lagrange

☐ Euklid-Primzahltest inklusive Fehlerwahrscheinlichkeit und Laufzeit kennen.

☐ Fermat-Primzahltest inklusive Fehlerwahrscheinlichkeit und Laufzeit kennen. Definitionen von Pseudoprimzahlbasis und Carmichael-Zahl kennen.

☐ Für den Miller-Rabin Primzahltest sollte es reichen, die Fehlerwahrscheinlichkeit und Laufzeit zu kennen.

☐ Wissen wie man Duplikate mithilfe einer Hashfunktion finden kann, inklusive Laufzeit, und wie man m wählen muss, damit es nur konstant viele Kollisionen gibt.

☐ Hase-Igel Algorithmus grob erklären können, inklusive Laufzeit.

Graphen-Algorithmen

☐ Beweis der NP-Vollständigkeit für das Long-Path Problem kennen.

☐ Den Monte-Carlo Long-Path Algorithmus schrittweise erklären können, inklusive dem DP-Algorithmus und allen dazugehörigen Beweisen.

☐ Definition von Netzwerk, Fluss, ganzzahlig, Flusswert kennen.

☐ Definition eines s-t-Schnitts und dessen Kapazität kennen und wissen dass $\mathrm{cap}(S,T) \geq \mathrm{val}(f)$.

☐ Maxflow-Mincut Theorem kennen.

☐ Definition des Restnetzwerkes kennen und zeichnen können.

☐ Definition von augmentierenden Pfaden kennen und diese im Restnetzwerk bestimmen können.

☐ Den Ford-Fulkerson Algorithmus erklären können inklusive Laufzeit, und damit einen maximalen Fluss und einen minimalen Schnitt bestimmen können.

☐ Satz von Menger, Bipartite Matchings und minimaler Kantenschnitt als Flussproblem umformulieren können.

☐ Probleme als Flussprobleme formulieren können.

☐ Definition von $\mu(G)$ und einer Kontraktion kennen und wissen, wie Kontraktionen $\mu(G)$ verändern (Lemma 3.20) und die Wahrscheinlichkeit kennen, dass $\mu(G)$ erhalten bleibt (Lemma 3.21). Für diese beiden Lemmas sollte man die Beweise kennen.

☐ Erfolgswahrscheinlichkeit und Fehlerreduktion von $\mathrm{Cut}(G)$ sollte man grob kennen.

- [ ] Bootstrapping sollte man grob erklären können, und die Laufzeit des Limit-Algorithmus kennen.

Geometrische Algorithmen

- [ ] Die Definition des kleinsten umschliessenden Kreises kennen, inklusive Eindeutigkeit.
- [ ] Wissen dass es immer ein Zertifikat Q mit höchstens 3 Punkten gibt (Lemma 3.26).
- [ ] “Schlechte” Algorithmen für den kleinsten umschliessenden Kreis inklusive Laufzeit kennen.
- [ ] Die clevere Version (bei der die Punkte verdoppelt werden) inklusive Laufzeit kennen.
- [ ] Den Beweis von Lemma 3.28 kennen.
- [ ] Definition kennen von: Liniensegment, konvexe Menge, konvexe Hülle, allgemeine Lage, Randkante.
- [ ] Jarvis-Wrap und Find-Next inklusive Laufzeit erklären können.
- [ ] Local Repair Algorithmus inklusive Laufzeit grob erklären können.
- [ ] Untere Schranke für die Laufzeit der konvexen Hülle kennen.

Code Expert

Für Flow Aufgaben:

- Stellt sicher, dass ihr die Aufgabe exakt verstanden habt. Eine kleine Änderung kann zu einem ganz anderen Netzwerk führen.
- Macht das Indexieren der Nodes immer gleich und zeichnet eine Skizze, wo ihr die Indices notiert.
- Schaut euch noch einmal einige Graph Modelling Tricks von A&D an wie z.B. Graph-Layering, Super-nodes, Kanten umdrehen, Logarithmus auf Gewichte anwenden

Für Wahrscheinlichkeits-DP Aufgaben:

- Schreibt euer DP Programm vorzugsweise rekursiv (=top-down) statt iterativ (=bottom-up), da es sein kann, dass die Berechnungsreihenfolge manchmal ziemlich schwierig ist.
- Falls ein Erwartungswert gesucht ist, dann sollte man ziemlich sicher Satz 2.32 verwenden.
- Falls eine Wahrscheinlichkeit gesucht ist, dann sollte man ziemlich sicher den Satz der totalen Wahrscheinlichkeit verwenden.

Some Jokes to survive Lernphase

A guy is terrified of flying because the probability of a bomb being on a plane is 1/1000. So, he decides to always pack a bomb in his own suitcase.
His reasoning? "The probability of *one* bomb on a plane is 1/1000, but the probability of *two* bombs is 1/1,000,000! Since I brought one, the flight is now much safer."

I surveyed many people who had played Russian roulette. Seems like the probability of dying is actually 0%.

What is the similarity between me and an experiment involving a biased coin with two tails? The probability of getting a head is zero.

If you didn’t like these jokes, please keep in mind that not all math puns are terrible. Just sum.