

Wir schon besucht haben und welche wir noch besuchen müssen. Da gibt es Datenschutzpunkte. Was ist eine Datenschutzpunkte? Ihr habt bestimmt ein paar Programmierer hier, oder? Keiner? Und der traut euch nur nicht zu melden. Okay, dann können wir uns das gemeinsam mal überlegen.

Zum Weggewischt. Am Anfang sind wir in dem Knoten A. So.

Wenn der Knoten A kein Suchknoten ist, dann gucken wir uns den nächsten Knoten an. So, der B und C. So, wir hängen das also hinten an den Knoten A dran.

Der Knoten A, den haben wir jetzt fertig, da sind wir durch. Gehen wir auf den nächsten Knoten B. Ist keine Lösung, sind wir durch. Gehen wir dann C, ist keine Lösung. Was machen wir als nächstes? Wir gucken uns die nächste Ebene an. Und wie finden wir die nächste Ebene? So.

Von der nächsten Buchstaben-Ebene. Also unten dann natürlich.

Was war die nächste und unten? Also Ebene, von der oberen Ebene geht auf die nächste. Weil die erste Ebene nicht so gut war. Aber wir wissen immer nur diese Verbindung, die durch die Striche dargestellt ist. Also woher wissen wir jetzt?

Wir gehen zum ersten Knoten zurück. Den habe ich aber gestrichen. Den weiß ich nicht mehr. War also nicht so eine gute Lösung, oder? So.

Wenn ich nach A gehe, dann merke ich mir gleich auch, was sind die nächsten Knoten dahinter, die ich mir merken müsste. A kommen B und C, das habe ich gemacht. A ist fertig. Es gibt keinen anderen Knoten mehr.

So, jetzt gucke ich mir den Knoten B an. So, wenn das der Knoten, das Ziel ist, ist es okay. Wenn das nicht das Ziel ist, dann muss ich mir merken, hey, wenn ich auf der Ebene nichts finde, dann gucke ich mir, muss ich mir die Nachfolger anschauen.

Also ich muss mir merken, dass D.

So, die Frage ist, wo merke ich mir diese Nachfolger? Die kann ich mir hier merken, okay. Wenn ich C fertig bin, dann wäre als nächstes D und E dran. Okay, so, ich schiebe die also hinten dran.

Jetzt kann ich B streichen, wenn es keinen Nachfolger, keine Lösung mehr gibt, komme zu C. Gucke C an, ist das mein Zielzustand? Wenn es das nicht ist, was mache ich dann?

Genau, ich gucke mal, was sind die Nachfolger.

Okay, dann kann ich C abarbeiten, bin fertig. Komme zu D. Was mache ich? Ich gucke, ob es ein Zielzustand ist. Wenn es kein Zielzustand ist, was mache ich dann? Genau, ich muss mir die Nachfolger merken, die würde ich hinten dran hängen und kann dann weitermachen.

So, ich nehme zu einem Knoten immer die Nachfolger und merke sie mir. Und ich habe gesagt, ich merke es, sie immer hinten, ich hänge sie einfach immer hinten dran. Jedes Prinzip.

Von Informatik und irgendwas. Ja, zum Beispiel bei dem Diagramm. Also es gibt ja auch Breiten und Tiefen-Diagramme.

Wenn man es aufteilen will in verschiedene Sektionen, zum Beispiel Migros hat jetzt Migros Bank, keine Ahnung, Migros Inno, Kiosk, dann wird das noch vertiefter gespalten, oder? Ja, aber das ist nur die Darstellung, oder? Die habe ich auch. Ich meine mir zu merken, wer als nächstes dran kommt.

Ja, aber geht es da immer so, dass ich die nächste Möglichkeit hinten dran hänge? Ihr denkt, glaube ich, viel zu kompliziert. Ich bin ein ganz einfacher Mensch. Wir machen ein ganz einfaches Prinzip. Wo habe ich das? Wenn ich Migros an die Kasse gehe, der nächste muss ich hinten anstellen, oder? Das ist das Prinzip.

Das Neue kommt hinten dran.

Das ist ein Prinzip, das weit verbreitet sind in der Informatik. Das nennt man eine Schlange, Queue.

Oder First In, First Out. Wer zuerst kommt, war zuerst, oder? Also wer zuerst kommt, steht zuerst in der Schlange. Und die anderen müssen sich hinten anstellen. Also wer zuerst kommt, kommt auch als erstes wieder raus.

Also First In, First Out ist ein allgemeines Prinzip in der Informatik, das fängt man hier an. Oder Queue, Englisch, oder Schlange.

Dieses Prinzip, ganz einfach, wir gucken es, das ist eine ganz einfache Art, den Baum zu durchlaufen. Wir gucken einen Knoten an, wenn es Ziel ist, sind wir fertig. Wenn es nicht, merken wir uns die Nachfolgeknoten, von denen hängen sie hinten dran. Und so laufen wir durch. Als einzige Datenstruktur, die wir brauchen, ist das hier. Um es gezielt durchzulaufen.

Einfach die Knoten sich immer merken und die neuen hinten dran hängen.

Gut, FIFO-Strategie, First In, First Out, Schlange, Queue, genau, ich habe alles auf der Folie. Gut.

Also wie würden wir mit Breitensuche hier durchgehen? Ich glaube, es ist relativ offensichtlich. Ihr könnt es mir aber trotzdem mal sagen.

Das ist es nicht. Dann wäre die Alternative, statt dass wir es nach unten geschoben hätten, hätten wir es nach links schieben können. Dann kommen wir nach C. Auch noch nicht, dann gehen wir, gucken wir, ob das für uns eine rechtsschiebende Lösung ist. Nein, das ist es nicht. Also muss ich zwei Schritte weitergehen.

Und wir sind relativ gut, uns noch einen, vielleicht zwei Schritte zu merken oder im Kopf vorauszudenken, außer wir sind geübt in so einem Beispiel. Und deshalb ist es gut, sich das immer zu merken und dann wieder, okay, wir gehen es ganz systematisch durch diesen Baum durch. Wie gesagt, das ist ein ganz allgemeines Prinzip und da gibt es auch einen Algorithmus dafür.

Weißt du, was ein Algorithmus ist oder? Was ist ein Algorithmus?

Genau, was ist das?

Keine Ahnung, ob es kommt. Immer wieder.

Es ist ein wiederholender Weg. Ich rede gerne ganz einfach mal. Was ist ein Algorithmus? Wir hören den Begriff ja oft. Was ist ein Algorithmus? Genau, eine Abfolge von Befehlen. Das ist ein Algorithmus. Manche nennen neuronales Netz auch ein Algorithmus. Das ist total, das ist kein Algorithmus.

Ein Algorithmus ist eine Abfolge von Befehlen. Und ein Kochrezept ist zum Beispiel auch ein Algorithmus. Ist eine Abfolge von Befehlen. Wenn das in der Programmiersprache steht, dann ist es eine Programmiersprache. Aber es ist ein Programm, das ist eine Implementierung von einem Algorithmus. Eine Abfolge von Befehlen ist ein Algorithmus.

Und hier habe ich einen Algorithmus für diese Breitensuche. Wir starten damit, dass unsere Liste von neuen Knoten leer ist. Ja, ich mache das nochmal mit dem Beispiel von.

Für alle Punkte, die wir schon haben. Ansonsten.

Fangen wir mit dem Knoten A an. Ist das ein Zielzustand? War nicht. Dann gehe ich hin und nehme B und C. Das ist das, was mit dem Event gemeint ist. Wir nehmen den neuen Knoten und hängen daran die Nachfolgeknoten.

So A war es, B und C. Und wenn es keine Nachfolge mehr gibt. Die Breitensuche von dem Rest.

Ja, dann müssen wir die Breitensuche von dem Rest machen. Und ansonsten, wenn es keine Lösung gibt, keinen Knoten mehr gibt, dann sind wir fertig. Breitensuche von dem Rest. Wir gehen einfach nach B weiter, um wieder den gleichen Algorithmus zu haben. Ist B der Zielzustand? Nein.

Dann nehmen wir den Nachfolgeknoten. Das waren B und E.

Das war das Prinzip.

Das ist eine Variante, wie wir da machen können. Und wir müssen uns aber immer merken, wie viele Knoten wir auf der Ebene haben. Es gibt noch eine Alternative, dass wir nicht immer alle. Wir ganz durchschauen können, das wäre die Tiefensuche. Tiefensuche habe ich im Prinzip schon erläutert.

Ich glaube, dass wir in der Breitensuche dort das Ziel finden.

Also wenn ein Knoten ein Zielzustand hat, hat man fertig. Ansonsten, wenn es kein Zielzustand ist, dann hat man zwei Möglichkeiten. Wenn es noch Nachfolgeknoten gibt, dann gehen wir dort in die Tiefe. Und suchen wir uns den ersten Nachfolgeknoten an. Und.

Ja, so geht das jetzt nicht.

Das habe ich auch grafisch dargestellt. Wenn wir links nochmal anschauen, dann hat Knoten und Nachfolgeknoten. Wir sind im Knoten A. Kein Zielzustand. Hat er Nachfolgeknoten, gehen wir in die Tiefe. Und den ersten Knoten schauen wir uns an. Also wir schauen uns den Knoten B an. So, machen wir das Gleiche. Ist es ein Zielzustand? Nein.

Dann müssen wir uns den Nachfolgeknoten anschauen.

Das

sind wir bei D. Jetzt hat, wenn das auch. Was können wir jetzt machen?

Dann nach E. Wir gehen einen Schritt zurück und schauen, ob wir dort Nachfolgeknoten haben. Das ist das, was wir Backtracking bezeichnen. Also zurück zur letzten Verzweigung, an der es noch Alternativen gibt. Gut. Das machen wir also E.

So, was machen wir dann? Wir gehen zurück zum B-Zug oder B. Genau, auf die zweite Ebene. Knoten, wo wir noch nicht alle Nachfolgeknoten angeschaut haben. Also was ist das nächste?

Aber der Kontext war bei dem.

Knoten haben, den wir noch nicht angeschaut haben. So, dann expandieren wir den. Kommt dann, geht das Gleiche. Wir gucken bei F den ersten.

So, wieder zurück. Jetzt haben wir bei F noch einen Nachfolgeknoten. Also wir gehen zu dem H. So, was passiert, wenn wir bei H nichts finden?

Kommen wir zu I.

Und wo haben wir noch Nachfolgeknoten? Nein. Bei A. Bei A. A hat noch eine offene Punkte. Also müssten wir es irgendwo gemerkt haben, dass A noch eine offene Punkt hat. Also von da gehen wir zu I. Und dann geht es genauso, wie wir es vorher auch gemacht haben. Und das kann man auch machen, wenn der Graph nicht vollständig ist.

Also hier, wir gehen durch bis zum tiefsten Punkt. Das ist kein Zielzustand. Dann kommt als nächstes die Nummer 5. Weil wir dabei drei noch Nachfolgezustände haben. Wenn das kein Zielzustand ist, dann haben wir bei 2 noch einen offenen Punkt. Und dann gehen wir auf die 6. Wenn das kein.

Ja, also wir müssen 2, 3, 4, 5, 8 dort. Gucken wir uns die 7 an. Ist kein Nachfolge. Kein Zielzustand. Gehen wir hoch zur 1 und gucken dort nach den Alternativen. Also es geht.

Jetzt müssen wir einfach einen Schritt zurückgehen.

Okay, hier habe ich das mal versucht, grafisch darzustellen. Bei dem anderen Bild auch. Also A ist der Zielzustand. Wir gehen zu B. Bei B gehen wir zu D. Und dann H. Dort suchen wir die Alternative.

Wenn bei D, wenn es kein bei H kein Zielzustand ist. Die Frage ist jetzt hier, wie vorhin, was wäre, wie in welcher Reihenfolge muss ich mir die Knoten merken, damit ich den Baum mit Tiefenzug folgen muss?

Gehen wir ein bisschen Zeit nach hinten. Also hier haben wir mal hinten dran gehängt. Und damit können wir uns einfach merken, was der nächste Punkt ist, den wir anschauen müssen. Ja.

Last in, First Out. Cool. Ja. Last in, First Out. Last in, First Out. LIFO-Prinzip.

Wo finden wir so ein Prinzip?

Wie so in einem Lift?

Okay, ansonsten müssen wir Platz machen. Ja. Okay. Ja, cool. Wo finden wir es noch?

Wenn ihr Aufgaben habt. Da kommt das nächste Ding, da kommt das nächste Ding. Das oberste könnt ihr wiederholen, weil es eben nur immer das oberste ist. Okay. Einen Stapel. Wir machen einen Stapel und arbeiten immer das oberste ab. Das ist das, was hinter diesem Prinzip steht. Wir können es uns mal anschauen.

Was ist das Beispiel, was man typischerweise macht? Das ist der Stapel. So A ist nicht das Ziel. Jetzt muss man merken, was sind die Nachfolgeknoten. Vorher habe ich sie hinten dran gehängt. Ja. Jetzt mache ich die volle Freigangslösung.

Oder umgekehrt, C und B. C muss ich mir als nächstes anschauen. Okay. So. Wenn B nicht das die Aufgabe, das Ziel ist, was mache ich als nächstes? Ich gucke mir die Nachfolger von B an. So wie ich es vorher auch gemacht habe.

So. Die Nachfolger von B sind D und E.

Ich lösche es. Und mache den drauf.

Genau, immer der oberste. D ist nicht mein Ziel. Also mache ich I und H in die obere.

Wir haben ein Ziel. Nein. H hat keine Nachfolger. Was passiert?

Das ist in dem Fall I. Okay. I ist eine Lösung.

Keine Nachfolger. Also nichts draufbauen. Was ist das nächste, was oben ist? Ist E. Meine Alternative, die ich bei B hatte. So, jetzt gucke ich mir E an. Wenn es keine Lösung ist, dann nehme ich die Nachfolger von E. Das wäre K. So, K und J.

Die wieder oben drauf.

Also immer vorne dran. Ja. Kann ich die Tiefensuche hochrealisieren.

Also bei Breitensuche hängen wir die Knoten immer hinten dran, die Nachfolgeknoten sind von Knoten, die keine Zielzustände waren. Und bei Tiefensuche hängen wir sie vorne dran. Also First in, First Out, Last in, First Out als die Prinzipien.

Und Last in, First Out

wird dann auch starten.

Und der Algorithmus ist auch relativ einfach. Statt dass wir vorne etwas hinten.

Dann können wir jetzt die neuen Knoten vorne dran hängen.

Also wir schauen. Ist erreicht, dann haben wir die Lösung gefunden. Ansonsten gucken wir neue Knoten, sind die Nachfolgeknoten. Und wir wenden die Tiefe auf den ersten Knoten von den Nachfolgeknoten an.

Und das ist das Prinzip, dass wir immer vorne weitermachen und den Rest von Nachfolgeknoten erst.

Gut, wollen wir uns mal gerade unser Beispiel anschauen von. Von dem Weg finden.

Also wenn wir diesen Graphen haben, den haben wir vorhin aufgebaut.

Um Breitensuche zu machen. Welche Städte würden wir zuerst anschauen, bevor man dort zum Ziel kommt? Bitte? Breitensuche.

Zürich und dann. Ja, Zürich dann. Dann Art Goldau. Wenn das da keine Zielzustände sind, gucken wir auch mal von Zürich nach Basel. Wo zum Weg finden. Und dann, wenn das nicht hilft, von Zürich nach Olten. Zürich nach Luzern. Und dann?

So, von Art Goldau nach Luzern. Wir lösen Luzern zweimal anschauen. Dummerweise haben wir da halt zweimal in unserem Graphen drin. So, wenn das sich zum Ziel führt, dann würden wir auf die nächste Ebene gehen. Wir gucken, ob der Weg über Solothurn uns zum Ziel führt. Okay. Dann gucken wir, Bern ist das schon ein Ziel. Dann Thun.

Dummerweise auch zweimal drin, weil wir Luzern zweimal drin haben. Also wir würden Ebene für Ebene anschauen. Und würden irgendwann nach Lausanne kommen. Und welche Strecke nach Lausanne würden wir finden?

Geht zweimal Lausanne. Oder dreimal. Welche würden wir finden? Genau, über Neuenburg. Weil, wenn wir die Ebene abgehen, ist das der erste Knoten, den wir, wenn wir ebenso vorgehen, der uns nach Lausanne führt.

Wir sind froh, wenn wir diesmal vorne eine Lösung finden. Wir wissen nicht, ob es die kürzeste ist. Aber es ist eine Lösung. Okay.

Weil wir gar nicht nach Kilometern geguckt haben. Wir haben einfach nur geschaut, gibt es einen Weg. Also wir machen uninformed Suche.

Wir müssen jetzt einfach zusammenzählen. Wir haben den Weg gefunden. Dann können wir sagen, das ist der Weg. Dann können wir die Kilometer zählen. Aber für unser Verfahren spielt es keine Rolle, wie wir auf die Einschätzung sprechen.

Man findet.

Gleichgewicht haben und dann gleiche Kosten haben.

Okay, ja. Ja, wir finden die Kosten einfach. Das andere, was.

Immer noch. Gucken wir weiter wäre, als überhaupt. Dann wäre es ein Zwischenschritt.

Das ist das, was ich machen will.

Schrittweise, ebenewise durchgehen. Das ist schön.

Den Weg nach Lausanne über Tiefzug.

Zuerst nach Zürich. Ja. Oder von Zürich nach Basel probieren. Basel geht ja nicht weiter, deswegen. Gut.

Bei Breitensuche, Breitensuche oder Tiefensuche haben wir meistens mehr Aufwand.

Ja, wir haben mehr Aufwand bei der Breitensuche. Wenn sich die Knoten sehr stark verzweigen, haben wir ziemlich viel Aufwand bei der Breitensuche. Wir müssten es also auch ziemlich viele Knoten als Zwischenstation merken.

Weil wir Pfade anschauen, die wir finden, die müssen tiefer. Ja. Zeitverlust und. Also auch generell Aufwand wäre höher, was die Dauer, die wir brauchen, eben auch, weil wir auch mehr Knoten merken müssen. Ja. Kurz und praktisch, was wir müssen merken. Das geht in die gleiche Richtung. Genau.

Und was könnte noch ein Problem sein?

Nein, eigentlich, sorry, ich habe die Frage falsch gestellt. Was könnte.

Was könnte ein Problem bei der Tiefensuche sein? So.

Also dieerste Möglichkeit wahrnehmen, nicht die zweite.

Das ist der Lenker, dass die zweite. Das könnte ein Problem sein, ja. Wenn es eben ein Eingriff, das eine Lösung durchbleibt in unmittelbar diesem Pass.

Genau, wenn ich eine Schleife habe, dann finde ich nie zu einer Lösung. Oder wenn die Wege ziemlich lang sind, dann gehe ich sehr lange in einen Pfad runter, der mich nicht zu einer Lösung führt. Und ich brauche lange über Backtracking, bis ich zu dieser Lösung finde.

Was könnte dann eine Alternative sein? Also Breitensuche ist ziemlich aufwendig.

Tiefensuche kann uns zum unendlichen Pfad führen. Beides sind Algorithmen. Tiefensuche und Breitensuche. Also wir brauchen einen anderen Algorithmus.

Wir sind doch froh, wenn wir eine haben. Wir wollen gar nicht mehr. Wir wollen jetzt. Ja, Sekunden habt ihr mehr zu tun. Kürzeste Pfad. Also kürzeste, wenigste. Wenigste Kosten soll das sein. Das gucken wir nachher.

Wenn man sagt, man macht die Breitensuche.

Das ist eine breitere Suche. Das ist doch die Idee relativ gut. Also wir machen Breitensuche, beschränken sie aber in der Tiefe. So.

Falsche Richtung.

Ich habe hier nochmal vorhin nachgefragt. Tiefensuche ist vollständig, Tiefensuche ist nicht unbedingt vollständig, wenn ich dann mal in eine Schleife gehe.

Also kürzeste Lösung. Tiefensuche nicht immer, weil der Pfad, den ich komme, könnte der längere sein. Deshalb habe ich das vorhin als Beispiel gebracht. Und die Rechenzeit ist deutlich höher mit der Zeit bei der Breitensuche. Wir brauchen bei der Tiefensuche viel weniger. So.

Also wir nehmen es vor, wir gucken erstmal auf der Ebene 1 in der Breite. Und wenn wir dort nichts finden, dann machen wir die Breitensuche nochmal von vorne, gehen aber nur bis zur Ebene 2. Wenn wir dort nichts finden, starten wir die Breitensuche wieder und gehen jetzt bis zur Ebene 3 und so weiter.

Also iterativ erhöhen wir immer die Tiefe der Breitensuche.

Die kürzeste Weg finden, das haben wir damit erreicht. Da war es die Tiefensuche nicht garantiert. Und wir laufen auch nicht in eine Schleife, die ganz tief ist, wenn es vorher über die Breitensuche einen Weg gegeben hätte.

Also würden wir einfach immer iterativ die Tiefensuche nehmen, oder? Nein?

Warum nicht? Weil es vielleicht auch andere Sachen gibt. Also bei der Breitensuche, wenn ich vielleicht nicht nur eine Sache suche, oder? Ja, wir suchen nur eine Sache. Davon gehen wir aus. Wir uns setzen. Was ist der Nachteil von dem?

Ja, es ist schon relativ aufwendig, weil wir wiederholen ja suchen. Wir wiederholen die ersten Ebenen wieder. Wenn wir sagen, wir erhöhen die Tiefe, dann machen wir das, was wir vorher schon gehabt haben, wieder neu. Ja, wir vermeiden einfach, dass es Endlosschleifen gibt. Dafür nehmen wir aber in Kauf, dass wir die Breitensuche relativ häufig anwenden. Und die ist ja aufwendig, wie wir gesagt haben.

Also ist das nicht unbedingt die beste Lösung.

Also Tiefensuche oder eine heuristische Suche. Wisst ihr noch, was eine heuristische ist? Faustregel, genau.

Heuristische Suche, ja, da müssen wir jetzt mehr Informationen haben, sozusagen können. Der nächste Schritt führt uns das wahrscheinlich zu einer Lösung oder nicht? Und wenn wir denken, ja, das könnte uns zu einer Lösung führen, dann können wir heuristische Suche anwenden.

Mit dem Graphen, den wir hatten bei der Entfernung zwischen Städten, wenn es ungefähr in die gleiche Himmelsrichtung geht, dann ist dieser Schritt zu bevorzugen, als wenn es in eine andere Himmelsrichtung geht. Wenn wir diese Heuristik anwenden, dann haben wir vielleicht eine Chance, schneller den Weg zu finden, als wenn wir einfach immer alle Nachfolgeknoten anschauen.

Gut, Breitensuche haben wir die uninformedierte Suche genannt, Tiefensuche auch, weil wir den ganzen Suchbaum immer durchlaufen müssen. Oft sind aber ganz einfach. Das ist auch nur eine sehr gute Lösung.

Und dann können wir gucken, ob wir mit weniger Aufwand eigentlich eine sehr gute Lösung finden und werden damit schon zufrieden. Und das ist das, was heuristische Suche macht. Wenig Aufwand, aber eine gute Lösung finden.

Und Heuristiken sind solche Strategien, die uns relativ schnell zu einer Lösung führen, also schneller als nur die uninformede Suche. Was wir dafür brauchen, ist aber irgendeine Funktion, die sagt, sind wir näher an unserem Ziel oder nicht?

Also wir müssen diesen Aufwand irgendwo mitberechnen. Und jetzt nochmal ein Vergleich. Wir hatten die uninformede Suche gehabt. Entweder wir nehmen die neuen Knoten, stecken sie am Ende hin. Das war das FIFO-Prinzip. Oder wir fügen uns am Anfang dran. Das war das LIFO-Prinzip.

Und wenn wir jetzt heuristische Suche machen, dann machen wir nicht strikt, okay, wir führen das am Ende an oder vorne an, sondern wir sortieren die Knoten so nach dem Prinzip, was führt uns wahrscheinlich näher zum Ziel.

Okay, und das heißt, wir müssen irgendwo die Entfernung zum Ziel abschätzen können. Und das ist das, was wir als Bewertungsfunktion bezeichnen. Wir wollen abschätzen können, ob wir näher zum Ziel kommen. Und das muss weniger aufwendig sein.

Also wenn wir dafür sehr viel. Vielleicht machen wir es.

Gucken wir uns das nächste Mal an, weil ich sehe gerade, es ist schon 12 Uhr. Wir machen nächstes Mal heuristische Suche und schauen, wie wir relativ effizient eine Lösung finden können. Und wahrscheinlich fangen wir dann auch noch mit einem anderen Thema an, Logik.

Die beiden Sachen sind grundlegend für die Informatik. Deshalb wollen wir uns gucken, was diese beiden Sachen sind.

Das hat wahrscheinlich der Manuel gesagt, oder? Aber was wollen wir wissen zur Prüfung?

Also die Prüfung wird eine Online-Prüfung sein. Ihr kennt ja Moodle. Dort kann man auch Quizzes machen. Wir werden die Prüfung als Quiz in Moodle machen. Teilweise mit geschlossenen Fragen, also wo man einfach ankreuzen kann.

Und teilweise mit offenen Fragen, sodass wir eine Aufgabe stellen. Also hier zum Beispiel könnte es sein, in welcher Reihenfolge werden die Knoten abgearbeitet? Und ihr müsst euch die Reihenfolge hinschreiben.

Wir erlauben euch, ein Spickzettel zu machen. Wir würden uns auch den Spickzettel vorher mal anschauen, wenn ihr sagt, passt das oder nicht, ist das erlaubt. Also der Spickzettel, haben wir gesagt, sind zwei Seiten.

Also ein Blatt, das auf beiden Seiten geschrieben werden kann. Es kann nicht zwei Blätter, sondern ein Blatt und zwei Seiten. Ja.

Also ihr solltet den Spickzettel machen. Hat einfach einen didaktischen Grund. Wenn ihr den Spickzettel selber macht, müsst ihr euch das mal anschauen und damit lernt ihr besser. Eigentlich uns geht es ums Lernen. Und deshalb haben wir gesagt, wir machen das so. Ihr generiert den Spickzettel selbst und dann dürft ihr ihn auch mitnehmen. Aber es kommen ja nicht nur Sachen von den Folien dran.

Sondern auch das, was wir hier erklärt haben. Ja. Also auch das, was wir drumherum erklären, ist Teil. Kann in der Prüfung dran kommen. Was sehe ich das mit dem Roboter zum Beispiel aus? Das steht auch irgendwo auf den Folien. Ja, aber im Prinzip schon.

Wenn wir jetzt, wenn wir eine Diskussion führen und es ist ein Thema gewesen, dann kann es sein, dass wir auch eine Frage dazu schreiben. Aber ja, der Großteil ist auf den Folien. Also ist klar, wir

wollen ja auch, dass ihr lernen könnt. Und deshalb stellen wir euch die meistens in der Regel die Sachen zur Verfügung.

Und es steht irgendwo auf den Folien.

Mit was? Das machen wir gleich. Ich habe noch einen Moment zur Prüfung. Also wir gucken auch in der letzten Unterrichtsstunde das nochmal gemeinsam an. Wir geben dort Informationen, nachdem wir alles behandelt haben, wie die Prüfung aussehen könnte, auf was ihr lernen müsst. Also ihr kriegt noch deutlich mehr Informationen.

Aber wenn ihr auch wisst, was überhaupt geprüft werden kann. Jetzt ist die Information relativ vage. Aber wir machen das auch noch detailliert. Wir geben euch auch Prüfungsfragen, Beispielfragen, die ihr vorher ausprobieren könnt. Damit ihr auch wisst, wie die Prüfung dann aussieht.

Wir können die Boden hantieren, sodass das keine Neuigkeiten für euch sind. Also ihr seid gut vorbereitet.

Ist es noch nicht kompliziert?

Ich meine, das ist schon vorhanden. Das passt einfach mal ins Kopieren auf. Also wenn ihr es noch nicht habt, kommt es in den nächsten Tagen. Also wir haben schon das Feedback gekriegt vom Prüfungssechspaar Jahr. Wir konnten Feedback geben. Also wenn ihr es noch nicht habt, kommt es in den nächsten Tagen. Da war hinten noch jemand, den frage ich.

War die gleiche. Okay. Gut, dann wünsche ich euch noch einen schönen Tag. Ciao.

Das ist eine schöne Uhr. Ja.