Methoden für den Entwurf von Algorithmen

- **Greedy Algorithmen:**
  - Löse ein einfaches Optimierungsproblem durch eine Folge „vernünftiger“ Entscheidungen.
  - Eine getroffene Entscheidung wird nie zurückgenommen.
- **Divide & Conquer**:
  - Zerlege ein zu lösendes Problem in eine Hierarchie einfacherer Teilprobleme.
  - Der **Rekursionsschritt**: Löse ein schwierigeres Teilproblem mit Hilfe einfacherer Teilprobleme.
- **Dynamische Programmierung:**
  - Zerlege ein zu lösendes Problem in eine Hierarchie einfacherer Teilprobleme.
  - **Rekursionsgleichungen**: Löse ein schwierigeres Teilproblem mit Hilfe einfacherer Teilprobleme.
    - Lösungen der Teilprobleme werden abgespeichert.
    - Die Lösung eines Teilproblems wird möglicherweise für die Lösung mehrerer anderer Teilprobleme benötigt.

Greedy-Algorithmen (1/2)

- Dijkstra's Algorithmus:
  - Die Entscheidung: Lege einen kürzesten Weg für einen neuen Knoten fest.
  - Die Heuristik: Der neue Knoten besitzt einen kürzesten S-Weg.
- Prims und Kruskals Algorithmus:
  - Die Entscheidung: Setze eine neue Kante ein.
  - Die Heuristik: Wähle eine kürzeste kreuzende Kante.
- Interval Scheduling für einen Prozessor:
  - Die Entscheidung: Wähle eine neue Aufgabe aus.
  - Die Heuristik?

- Achtung: Greedy Algorithmen versagen, wenn Aufgaben Gewichte erhalten.
- Was tun? Dynamische Programmierung!

Greedy-Algorithmen (2/2)

- Minimiere die maximale Verspätung (für einen Prozessor).
  - Die Entscheidung: Welche Aufgabe wird als nächste ausgeführt?
  - Die Heuristik?
- Huffman Codes:
  - Die Entscheidung: Welche beiden Buchstaben werden zu Geschwistern?
  - Die Heuristik: Die beiden Buchstaben geringster Häufigkeit.

Der **Korrektheitsbeweis** ist der wesentliche Schritt in der Analyse eines Greedy Algorithmus.

Divide & Conquer

Divide & Conquer Algorithmen:

- Binärsuche,
- Präorder und Postorder Traversierung von Bäumen,
- Tiefensuche,
- Quicksort und Mergesort,
- die schnelle Multiplikation ganzer Zahlen
- sowie die schnelle Multiplikation von Matrizen.

Jedesmal wird das Ausgangsproblem rekursiv gelöst, nachdem ähnliche, aber einfachere Probleme gelöst wurden.

Dynamische Programmierung

Die wesentlichen Schritte im Entwurf:

- Definiere die **Teilprobleme** und
- bestimme die **Rekursionsgleichungen:** Löse schwierigere Teilprobleme mit Hilfe bereits gelöster, einfacherer Teilprobleme.

Dynamische Programmierung: Der Orakel-Ansatz

Angenommen, ein Orakel beantwortet Fragen über eine optimale Lösung: Was ist unsere jeweils erste Frage?

- Für das gewichtete Intervall Scheduling: Wird die Aufgabe mit spätester Terminierungszeit ausgewählt oder nicht?
- Für kürzeste Wege-Probleme:
  - Der Algorithmus von Floyd: Wird Knoten n durchlaufen?
  - Bellman-Ford: Welche Kante wird als erste gewählt?
  - TSP: Welche Kante wird als erste gewählt?
- Für das paarweise Alignment der Strings u und v:
  Wird der letzte Buchstabe von u gegen den letzten Buchstaben von v oder gegen das Blank-Symbol aligniert?

Dynamische Programmierung: Gute Fragen

- Wann ist eine Frage gut?
  - Bei bekannter Antwort ist nur noch ein einfacheres Problem zu lösen.
- Aber wir kennen die Antwort nicht!
  - Wir machen unsere Fragen zu Teilproblemen
  - und lösen die Teilprobleme mit Hilfe der Rekursionsgleichungen.
- Achtung:
  - Der Algorithmus beginnt mit der Lösung der einfachsten Teilprobleme und arbeitet sich dann „hoch“.
  - Der Orakelansatz beginnt mit dem schwierigsten Teilproblem und arbeitet sich dann „herunter“.

Finde **alle** an das Orakel gestellten Aufgaben heraus: Dann sind **alle** Teilprobleme gefunden!

Teilprobleme und Rekursionsgleichungen

- Welche Teilprobleme und
- welche Rekursionsgleichungen haben wir für unsere verschiedenen Algorithmen benutzt?

World Series

Im Baseball werden die „Weltmeisterschaft“ zwischen zwei Teams A und B entschieden:
Das Team, das zuerst 4 Spiele gewonnen hat, wird Weltmeister.

- Diesmal werde die Weltmeisterschaft entschieden, wenn ein Team zuerst n Spiele gewonnen hat.
- Beide Teams sind gleichstark: Team A geinnt also ein bestimmtes Spiel mit Wahrscheinlichkeit $1/2$.
- **Berechne** $P(i, j)$ die Wahrscheinlichkeit, dass Team A Weltmeister wird, wenn Team A noch i Spiele gewinnen muss und B noch j Spiele gewinnen muss.