

1 potentielle Prüfungsfragen

1.1 Logik, Zahlen und Zahlensysteme

1. Wie funktioniert der Grey-Code?
2. Führen Sie die Multiplikation zweier Binärzahlen a, b vor ($16 < a, b < 30$)
3. Was ist die Binärdarstellung von $1/10$?
4. Wie sieht eine Floatingpoint-Zahl in IEEE 754 double precision aus?
5. Erklären Sie den Aufbau einer single-precision Floatingpoint-Zahl an einem guten Beispiel (dez. und bin. zeigen)
6. Was bedeutet shiften?
7. Wozu brauchte man Logarithmentafeln?
8. Wie funktioniert das Horner Schema (Beispiel) und was ist sein Vorteil? Hat es denn auch Nachteile?
9. Wie viele single-precision floating-point Zahlen gibt es mit dem gleichen Exponenten?
10. Welche ist die kleinste Zahl im IEEE 754 single precision Floatingpoint-Format, die größer ist als 100?
11. Stellen Sie die Wahrheitstabellen von AND, OR, NOT, XOR, NAND und NOR auf.
12. Wieviele Boole'sche Funktionen in 8 und in 9 Variablen gibt es, und warum?
13. Wie kann man AND, NOT und OR durch NAND ausdrücken?
14. Was ist das Dualitätsprinzip für Boole'sche Ausdrücke? Beispiel!
15. Vereinfachen Sie $a \wedge b \wedge c \vee b \wedge \neg c \vee a \wedge c \vee a \wedge b \wedge \neg c$
16. Welche logischen Operationen kennen Sie?
17. Geben Sie ein Beispiel einer Konjunktion von Disjunktionen aus 12 Binärvariablen.

1.2 Formale Sprachen, Automaten

1. Nennen Sie ein Beispiel für eine Grammatik in Backus-Naur-Form mit mindestens 6 Regeln.
2. Zeigen Sie: Es gibt überabzählbar viele Funktionen $f : \mathbb{N} \rightarrow \mathbb{N}$ warum?
3. gibt es mehr Funktionen $\mathbb{N} \rightarrow \mathbb{N}$ oder mehr Algorithmen?
4. Warum gibt es nur abzählbar unendlich viele Computerprogramme?
5. Was besagt die Church'sche These?
6. Was ist eine Turingmaschine? Wie funktioniert eine Turingmaschine?
7. Welche formale Sprache kann von einem endlichen Automaten nicht erkannt werden?
8. Welche formale Sprache kann von einem Kellerautomaten nicht erkannt werden?
9. Zeigen Sie: Jedes Computerprogramm kann mit einer einzigen While-Schleife auskommen.
10. Wie unterscheiden sich die GOTO- und die WHILE-Sprache?
11. Welche drei Arten von Maschinen erkennen welche drei Klassen formaler Sprachen?
12. Entwerfen Sie eine Turingmaschine, die die Sprache $a^n b^n c^n$ erkennt (4 Punkte)
13. Erklären Sie den Zusammenhang von Symbol, Alphabet, Wort und formaler Sprache.
14. Zeichnen Sie einen endlichen Automaten, der nur Zahlen in wissenschaftlicher Zahldarstellung akzeptiert wie -1.23432e22.
15. Woran erkennt man einen nichtdeterministischen endlichen Automaten?
16. Warum kann ein endlicher Automat keine Klammerausdrücke erkennen?
17. Was ist eine Taxonomie? Ein gutes Beispiel?

1.3 Programmiersprachen, Datentypen

1. Sind interpretierte oder kompilierte Programme schneller?
2. Warum divergiert die harmonische Reihe in C scheinbar nicht?
3. Zählen Sie einige Programmiersprachen auf (je 5 Sprachen 1 Punkt)
4. Was ist der Call Stack?
5. Was unterscheidet Klassen in C++ von structs in C?
6. Was ist eine Klassenhierarchie? Beispiel!
7. Welche vier Dinge braucht es für einen Algorithmus?
8. Wieso braucht ein Algorithmus einen Korrektheitsbeweis?
9. Erklären Sie den Unterschied zwischen dynamischem und statischem Linken.
10. Der Compiler meldet einen Linkerfehler. Was kann die Ursache sein, was können Sie tun?
11. Wann kommt es zu einem Overflow?
12. Wozu dienen Konstruktor und Destruktor?
13. Wie sind Effektivität und Effizienz definiert?
14. Zeichnen Sie ein Beispiel einer doppelt verketteten Liste auf.
15. Wie funktioniert PROLOG? (ev. Beispiel)
16. Was ist PostScript? (evtl. Beispiel)
17. Wie hängen sie zusammen: Kompilieren, Linken, Objectdatei, Library, Executable?
18. Was sind zusammengesetzte Datentypen?
19. Erläutern Sie kurz die Aussage: "Information = Daten + Semantik"
20. Setzen Sie fort: "Ein Graph besteht aus..."
21. Definieren Sie, was ein Baum ist.
22. Was ist der spannende Baum eines ungerichteten Graphen?
23. Zeichnen Sie zwei unterschiedliche isomorphe Graphen mit 8 Knoten und 20 Kanten.
24. Beschreiben Sie das Rebalancieren durch Rotieren (Skizze reicht).
25. Was ist ein DAG?
26. Was ist das Problem bei einer Mehrfachvererbung?
27. Was ist post mortem debugging?
28. Was ist der Zweck von Bytecode?
29. Was macht man mit dem gdb?
30. Beschreiben Sie die Arbeitsweise von make.
31. Welche zwei ganz unterschiedlichen Bedeutungen hat das Wort heap?
32. Was ist eine rekursive Funktion? Geben Sie ein sinnvolles Beispiel!
33. Wann setzt man call by reference ein, wann call by value.
34. Warum ist C keine objektorientierte Sprache?
35. Kann man objektorientiertes und strukturiertes Programmieren kombinieren?
36. Was ist der Unterschied zwischen Pointer(C++) und Referenz(Java)?
37. Geben Sie ein Beispiel für ein Programm, das mit stack overflow abbricht (Pseudocode).
38. Geben Sie ein Beispiel für Programmcode, der syntaktisch korrekt ist, nicht aber semantisch.

1.4 Algorithmen, Komplexität und O-Notation

1. Wieviele Teilmengen einer n -elementigen Menge gibt es, und warum?
2. Wenn sich n Menschen die Hände schütteln, wie oft werden Hände geschüttelt?
3. Wenn sich 100 Menschen die Hände schütteln, wie oft werden Hände geschüttelt?
4. Was ist die Bedeutung der Fibonaccifolge?
5. Welche Operationen gibt es auf doppelt verketteten Listen, welchen Aufwand haben sie?
6. Wie funktioniert die binäre Suche auf einem sortierten Array?
7. Wie funktioniert die binäre Suche auf einer sortierten/unsortierten doppelt verketteten Liste?
8. Wie lange dauert das Finden eines Elementes in einem sortierten Array der Länge n ?
9. Was bringt es, Binärbäume zu balancieren?
10. Welchen Aufwand haben Einfügen, Suchen, und Löschen in Arrays, doppelt verketteten Listen und balancierten Binärbäumen.
11. Geben Sie die mathematische Definition der Landau-Notation $O(n)$.
12. Wie funktioniert Insertion-Sort, und welchen Aufwand hat es?
13. Was ist eine Permutation? Wieviele davon gibt es bei n Elementen?
14. Was bedeutet "Matrizenmultiplikation ist $O(n^3)$ ", und warum ist das so?
15. Wie verhalten sich die Mengen $O(\log_{10} n)$ und $O(\log_2 n)$ zueinander?
16. Welche Funktionsklasse ist größer, $O(\log n)$ oder $O(\sqrt{n})$?
17. Wie viele Teilmengen einer n -elementigen Menge gibt es, und wie sieht man das ein?
18. Wozu braucht man Bucketsort?
19. Führen Sie am Beispiel vor, wie Mergesort ein Array von 17 Werten sortiert. Zeigen Sie dazu das Array nach jedem Merge-Schritt.
20. Wozu braucht man bei Heapsort den Heap?
21. Beschreiben Sie, was die Funktion `heapify-down` macht.
22. Erklären Sie Heapsort. Ersatzweise: Erklären Sie `heapifyUp` und `heapifyDown`.
23. Kann man in linearer Zeit sortieren?
24. Denken Sie sich 20 Zufallszahlen aus und zeigen Sie, wie Mergesort sie sortiert.
25. Wie funktioniert Quicksort, und welche Konsequenz hat die Wahl des Pivot-Elements?
26. Welche Strategie bei der Wahl des Pivot-Elementes kann dazu führen, dass Quicksort ein sortiertes Array in $O(n^2)$ sortiert?
27. Setzen Sie fort: $O(n)$ ist die Menge...
28. Welchen Aufwand hat die Addition zweier Binärzahlen?
29. Warum funktioniert der GgT-Algorithmus von Euklid?
30. Warum ist die Bestimmung der Fläche der Antarktis ein offenes Problem?
31. Ist $O(\log_{10} n) = O(\log_2 n) = O(\ln 2n)$?
32. Sie haben ein sortiertes Array von 14 Elementen. Welche Indizes werden bei Durchlaufen in Preorder, Inorder, Postorder benutzt?
33. Ist $f(n) = n^2$ Element von $O(n^3)$?
34. Ist $f(n) = n^3 + n^2$ Element von $O(n^3)$?
35. Wie sind NP-vollständige Probleme definiert?
36. Was bedeutet die polynomiale Reduktion eines Problems auf ein anderes?
37. Beschreiben Sie das 3-COLOR-Problem.
38. Beschreiben Sie das Rucksack-Problem.
39. Wie kann man das 3-COLOR-Problem auf SAT reduzieren?
40. Was wäre zu tun, um SAT auf 3-COLOR zu reduzieren?

1.5 Allerlei aus technischer und praktischer Informatik

1. Was ist der entscheidende Unterschied zwischen Internet und analogem Telefonnetz?
2. Beweisen Sie, dass das Halteproblem nicht gelöst werden kann. (4 Punkte)
3. Wie ist IPv6 aufgebaut?
4. Geben Sie eine kurze Erklärung der von-Neumann-Architektur.
5. Was ist eine UUID?
6. Erklären Sie den Begriff Sampling.
7. Beschreiben Sie eine Situation, in der Prefetching hilfreich ist.
8. Erklären Sie den Unterschied zwischen den Standards ASCII, Unicode und UTF.
9. Was ist ein Cache-Miss?
10. Was bedeutet Refactoring?
11. Erläutern Sie: "Was nicht getestet ist, funktioniert nicht."
12. Was ist der Unterschied zwischen Unicode und UTF?
13. Wie funktioniert UTF-16? Wie funktioniert UTF-8?
14. Was ist der Unterschied zwischen Latenz und Durchsatz eines Prozessors? Was ist größer, was kleiner. Kann beides gleich sein?
15. Was sind Mikrobefehle, und wozu werden sie gebraucht?
16. was bedeutet branch prediction?
17. Was ist prefetching, was eine cache line?
18. Was ist der Unterschied zwischen Multithreading und Multitasking?
19. Was ist der Unterschied zwischen einer race condition und einem deadlock?
20. Welche sieben Schichten hat das OSI-Modell?
21. Auf welcher OSI-Schicht spielen Ports eine Rolle?
22. Welche Standard-Ports kennen Sie?
23. Was machen Informatiker mit einer Shell?
24. Wann braucht man einen DOI?
25. Was war das Mailüflter!?
26. Wie hängen die Betriebssysteme VMS, Windows NT und Windows 2008 zusammen?
27. Funktionieren Computer deterministisch? (Pro/Kontra)
28. Warum ist analog genauer als digital? Warum ist digital genauer als analog?
29. Beschreiben Sie möglichst präzise den Unterschied zwischen analog und digital.
30. Diskutieren Sie die Frage, ob Informatik ohne Strom einen Sinn haben kann.

2 unwahrscheinliche Prüfungsfragen

...diese Themen wurden im WS 2013 nicht so detailliert betrachtet wie in den Jahren davor

1. Was sind wichtige Unterschiede zwischen C++ und Java?
2. Warum divergiert die harmonische Reihe?
3. Ihnen fällt zu einem diskreten Problem trotz viel Grübeleien nichts Besseres ein, als alle Lösungsmöglichkeiten durchzuprobieren. Was vermuten Sie, wie gehen Sie vor?
4. Was finden Sie auf www.python.org, wann brauchen Sie es?
5. Wann brauchen Sie www.antlr.org?
6. Archäologen finden in der Zukunft eine XML-Datei. Lässt sie sich einfach oder schwierig entschlüsseln? 2 Beispiele.
7. Wozu dienen die Programme `java` und `javac`?
8. Was ist der Unterschied zwischen Median und arithmetischem Mittel? Beispiel!
9. Was ist besser, MS Word oder LaTeX?
10. Welche Phasen hat das Wasserfallmodell?
11. Nennen Sie drei wichtige Eigenschaften (Vorteile) der agilen Software-Entwicklung.
12. Was ist ein Parse Tree? (Einf. Beispiel)
13. Was ist das kartesische Produkt von Mengen?
14. Ein Algorithmus betrachtet n Elemente, sortiert 5% aus, betrachtet die verbleibenden, sortiert davon wieder 5% aus, usw., bis schließlich ein letztes übrig bleibt. Welche Komplexität hat er?
15. Welche Daten müssen generell bei einem context switch ausgewechselt werden?
16. Erklären Sie Mutex und Semaphore?
17. Wie vermeidet man ein busy wait?
18. Was ist eine Methodentabelle, wie könnte man sie in C realisieren?
19. Welche Informationen enthält der Runtime-Stack eines C-Programms?
20. Kann C++ mehr als C?
21. Was ist eine virtuelle Funktion?
22. Was ist der Unterschied zwischen Überladen (overloading) und Überschreiben (overriding) von Funktionen?
23. Was ist eine allgemeine Schutzverletzung?
24. Wozu haben verkettete Listen einen Sentinel?
25. Diskutieren Sie Vor- und Nachteile von new/delete versus garbage collection.
26. Welche structs braucht man für gerichtete Graphen?
27. Warum sollte man in seinem C-Programm viele Assertions verwenden?
28. Was macht bei CORBA der ORB?
29. Beschreiben Sie das Wasserfall-Modell.
30. Welchen Vorteil hat das V-Modell gegenüber dem Wasserfall-Modell?
31. Welchen Nachteil hat das V-Modell?
32. Welche wichtige Eigenschaft unterscheidet Komposition von Aggregation?
33. Welche Arten von UML-Diagrammen gibt es?