

[Dashboard](#) / [Meine Kurse](#) / [Computer Science and Digital Communications - Bach...](#) / [CSDC23BB](#) / [Sommersemester 2021 \(CSDC23BB\)](#)
 / [ALDA CSDC23BB SS2021](#) / Prüfung / [Prüfung 13.9.2021](#) / [Vorschau](#)

Sie können diesen Test in der Vorschau ansehen. Wäre dies ein realer Versuch, würde dies abgeblockt, weil:

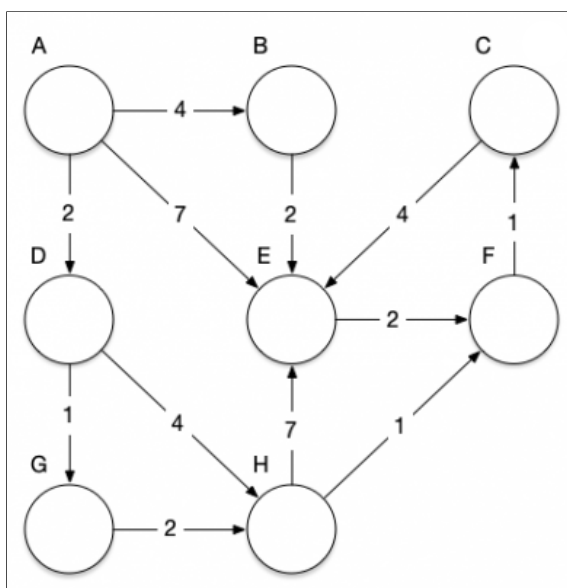
Dieser Test steht zur Zeit nicht zur Verfügung.

Frage 1

Bisher nicht beantwortet

Erreichbare Punkte: 10,00

Suchen Sie mittels Dijkstra Algorithmus den kürzesten Weg vom Knoten **D** nach **E**. Tragen Sie in die Knoten die Werte nach dem letzten Schritt ein.



0 1 2 3 4 5 6 7 8 9 10 11 unendlich

Frage 2

Bisher nicht beantwortet

Erreichbare Punkte: 5,00

Geben Sie die Reihenfolge an, in welcher der Dijkstra-Algorithmus die Knoten in der letzten Frage abarbeitet. Knoten, die eventuell nicht abgearbeitet werden, stellen sie bitte ans Ende der Reihenfolge

G E C F H D

Frage **3**

Bisher nicht beantwortet

Erreichbare Punkte: 5,00

Geben Sie den kürzesten Weg an, den Sie in der letzten Frage gefunden haben (in der Form a,b,c,d,e,f durch Beistriche getrennt):

Antwort:

Frage **4**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Welche der folgenden Algorithmen eignen sich zum Auffinden des kürzesten Weges?

Wählen Sie eine oder mehrere Antworten:

- ☐ Bellman-Ford Algorithmus
- ☐ Ford-Fulkerson Algorithmus
- ☐ Dijkstras Algorithmus
- ☐ DFS Algorithmus

Frage **5**

Bisher nicht beantwortet

Erreichbare Punkte: 1,00

Welche der folgenden Aussagen treffen auf den Bellman-Ford Algorithmus zu?

Wählen Sie eine oder mehrere Antworten:

- ☐ Im i.ten Durchlauf werden alle Pfade der Länge $i - 1$ untersucht.
- ☐ Funktioniert auch für negative Kantengewichte.
- ☐ Basiert auf Greedy Prinzip.
- ☐ Für die Umsetzung wird eine Queue verwendet.

Frage **6**

Bisher nicht beantwortet

Erreichbare Punkte: 1,00

Welche der folgenden Aussagen treffen auf den Dijkstras Algorithmus zu?

Wählen Sie eine oder mehrere Antworten:

- ☐ Basiert auf dem Divide & Conquer Prinzip.
- ☐ Zur Umsetzung wird eine Priority Queue verwendet.
- ☐ Funktioniert nur für nichtnegative Gewichte.
- ☐ Basiert auf Greedy Prinzip.

Frage 7

Bisher nicht beantwortet

Erreichbare Punkte: 3,00

Welche Aussage trifft auf die asymptotische Schranke zu?

Wählen Sie eine oder mehrere Antworten:

- ☐ Konstante Faktoren werden unterdrückt
- ☐ Sind eine generelle Methode um das Wachstum von Funktionen abzuschätzen.
- ☐ Gibt an, wie sich eine Funktion $f(n)$ zu $g(n)$ für $n > 0$ verhält
- ☐ Kann auf die Laufzeit oder den Speicherplatz angewendet werden.

Frage 8

Bisher nicht beantwortet

Erreichbare Punkte: 5,00

Ordnen Sie folgenden Funktionen nach der Dominanz (Wachstumsrate) beginnend mit der asymptotisch am schwächsten wachsenden Laufzeitfunktion

$30n^2$	$100n$	3^n	5^{2n}	$\frac{30n^4}{n}$	n^n
---------	--------	-------	----------	-------------------	-------

Frage 9

Bisher nicht beantwortet

Erreichbare Punkte: 5,00

Geben Sie die Größenordnung der Laufzeit für die folgende Funktion an:

```
// Laufzeit in Abhängigkeit von n und m
func(n, m)
1: for i = 1 to n do
2:   j = m;
3:   while j > 1 do
4:     j = j / 10;
```

A ▼	B	I									U		x_2	x^2					

Frage **10**

Bisher nicht beantwortet

Erreichbare Punkte: 5,00

Geben Sie die Größenordnung der Laufzeit für die folgende Funktion an:

```
Funktion1( n ) // Laufzeit in Abhängigkeit von n
1: if ( n <= 1 ) return 1
2: return Funktion1(n - 1) + Funktion1(n - 1)
```

A ▾	B	I											x_2	x^2					

Frage **11**

Bisher nicht beantwortet

Erreichbare Punkte: 5,00

Geben Sie die Größenordnung der Laufzeit für die folgende Funktion an:

```
Funktion2(n, m) // Laufzeit in Abhängigkeit von n und m
1: for i = 1 to n
2:   for j = 1 to m
3:     a[1] = a[i] * a[j]
4:     j = j + 2
5:   i = i + 2
```

A ▾	B	I											x_2	x^2					

Frage **12**

Bisher nicht beantwortet

Erreichbare Punkte: 10,00

Geben Sie einen Algorithmus in Form von Pseudo-Code an, der die folgende Spezifikation erfüllt.

Überprüfen Sie ob die Vorbedingung erfüllt ist:!

- Eingabe: $G(V, E, w)$... Graph mit Knoten und Kanten,
s ... Startknoten, t ... Zielknoten
- Vorbedingung: s, t ... müssen im Graph enthalten sein
- Ausgabe: L ... Liste aller Knoten, entlang eines beliebigen Weges von s zu Knoten t bei dem alle Kanten entgegen ihrer Richtung verwendet werden (Hinweis: siehe DFS)
- Nachbedingung: Pfad enthält t, wenn es einen Weg gibt

A ▼

B

I

U

~~S~~

x_2

x^2

Frage **13**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Eine Queue arbeitet nach dem folgenden Prinzip:

Wählen Sie eine Antwort:

- ☐ LIFO
- ☐ FILO
- ☐ FIFO

Frage **14**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Welche Funktion übernimmt der folgende Pseudocode im Zusammenhang mit doppelt verketteten Listen?

```
1: falls L1 == NULL dann {  
2:   L1 = L2;  
3: } sonst falls L2 ≠ NULL dann {  
4:   last = L2.Vorgänger;  
5:   L1.Vorgänger.Nachfolger = L2;  
6:   last.Nachfolger = L1;  
7:   L2.Vorgänger = L1.Vorgänger  
8:   L1.Vorgänger = last;  
9: }  
10: n1 = n1 + n2;  
11: retourniere L1;
```

Wählen Sie eine Antwort:

- ☐ Suche(x,L)
- ☐ Zugriff(p,L)
- ☐ Verkette(L1,L2)
- ☐ Entferne(p,L)
- ☐ Füge_ein(x,p,L)

Frage **15**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Ein Stack arbeitet nach dem folgenden Prinzip:

Wählen Sie eine Antwort:

- ☐ LIFO
- ☐ LILO
- ☐ FIFO

Frage **16**

Bisher nicht beantwortet

Erreichbare Punkte: 1,00

Welchem Entwurfsprinzip für Algorithmen folgt Merge Sort?

Antwort:

Frage **17**

Bisher nicht beantwortet

Erreichbare Punkte: 1,00

Ordnen Sie die Laufzeiten den entsprechenden Sortialgorithmen zu.

Merge Sort	Auswählen ...
Quick Sort	Auswählen ...
Insertion Sort	Auswählen ...
Selection Sort	Auswählen ...

Frage **18**

Bisher nicht beantwortet

Erreichbare Punkte: 4,00

Gegeben sei die unsortierte Folge (Array)

$A = 6, 4, 1, 8, 3, 2, 7, 5, 9$

In welcher Reihung befindet sich diese Folge bei Insertion Sort nach dem ersten, vollständigen Schleifendurchlauf der äußeren Schleife?

Insertion-Sort(A):

```
1: VAR key, i, j: INTEGER
2: für j = 2 ... n {
3: key = A[j];
4: //füge A[j] in sortierte Folge A[1],...,A[j-1] ein
5: i = j - 1;
6: solange i > 0 und A[i] > key {
7: A[i+1] = A[i];
8: i = i - 1;
9: }
10: A[i+1] = key;
11: }
```

Wählen Sie eine Antwort:

- ☐ 4, 1, 6, 8, 3, 2, 7, 5, 9
- ☐ 1, 4, 6, 8, 3, 2, 7, 5, 9
- ☐ 4, 6, 1, 8, 3, 2, 7, 5, 9
- ☐ 6, 4, 1, 8, 3, 2, 7, 5, 9

Frage **19**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Wie werden AVL Bäume balanciert?

Wählen Sie eine oder mehrere Antworten:

- ☐ Balanciert wird nach dem Einfügen eines Elements.
- ☐ Ein wesentlicher Faktor ist die Bestimmung des Gleichgewichtskoeffizienten.
- ☐ Mittels Devide & Conquer Methode.
- ☐ Durch Rotationen nach links und rechts.
- ☐ Durch Aufteilung in zwei Teilbäume.

Frage **20**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Sind B-Bäume immer balanciert?

Wählen Sie eine Antwort:

- ☐ Ja
- ☐ Nein

Frage **21**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Der unten stehende Algorithmus "XYZ" wird bei binären Bäumen wofür eingesetzt?

```
void XYZ (node *start)
{
    if (start != NULL) {
        XYZ(start->left);
        printf("%d ", start->key);
        XYZ(start->right);
    }
}
```

Wählen Sie eine Antwort:

- ☐ Suche von Elementen
- ☐ PreOrder Ausgabe
- ☐ LevelOrder Ausgabe
- ☐ InOrder Ausgabe
- ☐ PostOrder Ausgabe

Frage **22**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Welche Aufgabe übernimmt der folgende Algorithmus in einem binären Baum?

```
node * do(node *n, int key) {  
    if (n == NULL) {  
        return NULL;  
    }  
    if (key == n->key) {  
        return n;  
    }  
    if (key < n->key) {  
        return do(n->left, key);  
    }  
    else {  
        return do(n->right, key);  
    }  
}
```

Wählen Sie eine Antwort:

- ☐ Baumtiefe ermitteln
- ☐ Einfügen
- ☐ Suchen
- ☐ Löschen

Frage **23**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Wenn die Elemente {1,2,3,4,5,6,7,8,9,10,...} (aufsteigend sortierte Zahlen) in einen binären Baum, genau in dieser Reihenfolge, eingefügt werden, bekommt der Baum die Form einer ...

Antwort:

Frage 24

Bisher nicht beantwortet

Erreichbare Punkte: 10,00

Sie empfangen Daten von einer Applikation in unregelmäßigen Abständen, die sie zwischenspeichern und anschließend verarbeiten. Sobald die Daten verarbeitet wurden, leiten Sie die Daten an eine weitere Anwendung zur Ablage weiter.

Welche Datenstruktur oder Datenstrukturen würden sie zum Zwischenspeichern verwenden? Begründen Sie Ihre Antwort.

A ▼

B

I

U

x_2

x^2

Frage 25

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Welche der folgenden Aussagen treffen auf Datenstrukturen zu?

Wählen Sie eine oder mehrere Antworten:

- ☐ Beispiele für Datenstrukturen sind Sets, Heaps, Listen, Graphen und Bäume.
- ☐ Datenstrukturen sind wiederverwendbar weil sie dazu tendieren modular und kontextabhängig zu sein.
- ☐ Datenstrukturen ermöglichen eine verständlichere Betrachtungsweise der Daten.
- ☐ Datenstrukturen organisieren Daten in einer Weise, die Algorithmen effizienter macht.
- ☐ Datenstrukturen sind immer auch verbunden mit den Operationen bzw. Aktionen die man mit ihnen durchführen kann.

Frage 26

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Welche der folgenden Fragestellungen sind nicht relevant bei der Analyse von Algorithmen?

Wählen Sie eine oder mehrere Antworten:

- ☐ Kann der Algorithmus effizient in der Programmiersprache implementiert werden?
- ☐ Wie schnell ist der Algorithmus?
- ☐ Ist der Algorithmus korrekt?
- ☐ Wieviel Speicherplatz benötigt der Algorithmus?

Frage **27**

Bisher nicht beantwortet

Erreichbare Punkte: 2,00

Was wird unter dem Begriff Hashing verstanden?

Wählen Sie eine Antwort:

- ☐ Der Zugriff auf eine verkettete Liste im Falle von Kollisionen.
- ☐ Bezeichnet das Bucket, in welches das Element gehört.
- ☐ Die Übergabe des Schlüssels eines Elements an die Hashfunktion.
- ☐ Die dynamische Allokation von Speicher in der Hashtabelle.

Frage **28**

Bisher nicht beantwortet

Erreichbare Punkte: 5,00

Der nachfolgende Algorithmus ist ein Vertreter welcher Lösungsstrategie/Klasse?

```
Algorithmus: Platziere(i)

for h = 1..8 do
  if feld in Zeile i, Spalte h nicht bedroht
  then
    Setze Dame auf i, h;
    if Brett voll /* i = 8 */
    then
      Gib Konfiguration aus;
    else
      Platziere(i+1);
    fi
  fi
od
```

Antwort: