Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Datenstrukturen

186.815 Algorithmen und Datenstrukturen 2 VU 3.0

Übungstest SS 2012

28. Juni 2012

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname:		Vorname:	
Matrikelnummer:		Studienkennzahl:	
Anzahl abgegebener Zusatzblätter:		Unterschrift:	

Legen Sie während der Prüfung Ihren Ausweis für Studierende vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden. Benutzen Sie bitte dokumentenechte Schreibgeräte (keine Bleistifte)!

Die Verwendung von Taschenrechner, Mobiltelefonen, PDAs, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	18	20	12	50
Erreichte Punkte:				

Viel Erfolg!

Aufgabe 1.A: Skiplisten **(18 Punkte)**

a) (6 Punkte)

Fügen Sie in eine anfangs leere randomisierte Skipliste S die folgenden Elemente gemäß ihrer zugehörigen Höhe in der vorgegebenen Reihenfolge ein:

Schlüssel	10	37	25	52	6	48	17	2	12
Höhe	0	1	0	0	0	1	2	3	1

Zeichnen Sie die Skipliste S (nur das Endresultat).

b) (4 Punkte)

Suchen Sie nach der Zahl 28. Wie viele Schlüsselvergleiche waren hierfür notwendig? (Vergleiche mit ∞ zählen mit.)

c) (4 Punkte)

Suchen Sie nach der Zahl 48. Wie viele Schlüsselvergleiche waren hierfür notwendig? (Vergleiche mit ∞ zählen mit.)

d) (4 Punkte)

Es wird ein neues Element in eine randomisierte Skipliste eingefügt, wobei die Münzwurf-Methode aus der Vorlesung bzw. aus dem Skriptum verwendet wird. Wie groß ist die Wahrscheinlichkeit, dass der Container des neuen Elements die Höhe 4 annehmen wird?

Aufgabe 2.A: Approximationsalgorithmen (20 Punkte)

Bei der **Best-Fit-Heuristik** für das *Bin-Packing-Problem* wird jeder Gegenstand nicht in die ehestmögliche Kiste gelegt sondern in die Kiste, die der Gegenstand am besten ausfüllt, d.h., dort wo am wenigsten Platz überbleibt.

Der folgende Algorithmus **Best-Fit(Gegenstände** $1, \ldots, N$**)** soll genau diese Idee verwirklichen, wobei K die Größe jeder Kiste, f_j den noch freien Platz in Kiste j und w_i die Größe von Gegenstand i angibt. Sie können davon ausgehen, dass $w_i \leq K$, $\forall i = 1, \ldots, N$, und $N > 0$ gilt.

Best-Fit(Gegenstände $1, \ldots, N$**)**

Eingabe: Gegenstände $1, \ldots, N$

Ausgabe: m Kisten, die die Gegenstände $1, \ldots, N$ beinhalten

```
m = 1; f_m = K;
für i = 1, ..., N {
    bestbin = 0; bestplatz = ∞; gefunden = false;
    für j = 1, ..., m {
        restplatz = f_j − w_i;
        falls restplatz ≥ 0 ∧ restplatz < bestplatz dann {
            bestplatz = restplatz;
            bestbin = j;
            gefunden = true;
        }
    }
    falls gefunden dann {
        packe Gegenstand i in Kiste bestbin;
        f_bestbin = f_bestbin − w_i;
    } sonst {
        m = m + 1;
        packe Gegenstand i in Kiste m;
        f_m = K − w_i;
    }
}
```

a) (8 Punkte) Führen Sie den Algorithmus **Best-Fit** für $K = 8$ auf die folgende Eingabereihenfolge aus und geben Sie an, welche Kiste welchen Gegenstand am Ende enthält:

Gegenstand i	1	2	3	4	5	6	7
Größe w_i	4	5	6	2	4	1	2

b) (4 Punkte) Welche (asymptotisch relative) Approximationsgüte ϵ gilt für **Best-Fit** jedenfalls? Begründen Sie Ihre Antwort.

c) (8 Punkte) Zeigen Sie mit einem Beispiel (mit mindestens 6 Gegenständen), dass dieses Verfahren nicht immer eine optimale Lösung liefert. Was können Sie aus *Ihrem Beispiel* über die Approximationsgüte schließen?

Aufgabe 3.A: Theorie **(12 Punkte)**

a) Gegeben sei ein dynamisches Array A, in dem Elemente eingefügt werden können. Wenn A noch nicht voll ist, wird ein neues Element an eine beliebige freie Position eingefügt. Wenn A voll ist, wird seine Größe verdoppelt, um das neue Element aufzunehmen. Dabei kann es zu Umplatzierungen im Speicher kommen, was einen entsprechenden Aufwand verursacht.

- (2 Punkte)

  Geben Sie die **Worst-Case Laufzeit** für eine Einfüge-Operation in O-Notation an.

- (4 Punkte)

  Nehmen Sie an, Sie würden die Potenzialmethode für eine amortisierte Analyse auf dieses Problem anwenden. Welche Kenngröße kann für die Potenzialberechnung herangezogen werden?

- (3 Punkte)

  Geben Sie die **amortisierte Laufzeit** für eine Sequenz von n aufeinanderfolgenden Einfüge-Operationen in O-Notation an.

b) (3 Punkte)

Geben Sie in O-Notation an, wie groß die Höhe eines Fibonacci-Heaps maximal werden kann, der n Elemente enthält.

Technische Universität Wien
Institut für Computergraphik und Algorithmen
Arbeitsbereich für Algorithmen und Datenstrukturen

186.815 Algorithmen und Datenstrukturen 2 VU 3.0

Übungstest SS 2012

28. Juni 2012

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: Vorname:

Matrikelnummer: Studienkennzahl:

Anzahl abgegebener Zusatzblätter: Unterschrift:

Legen Sie während der Prüfung Ihren Ausweis für Studierende vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden. Benutzen Sie bitte dokumentenechte Schreibgeräte (keine Bleistifte)!

Die Verwendung von Taschenrechner, Mobiltelefonen, PDAs, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

	B1:	B2:	B3:	Summe:
Erreichbare Punkte:	20	18	12	50
Erreichte Punkte:				

Viel Glück!

Aufgabe 1.B: Approximationsalgorithmen (20 Punkte)

Bei der **Best-Fit-Heuristik** für das *Bin-Packing-Problem* wird jeder Gegenstand nicht in die ehestmögliche Kiste gelegt sondern in die Kiste, die der Gegenstand am besten ausfüllt, d.h., dort wo am wenigsten Platz überbleibt.

Der folgende Algorithmus **Best-Fit(Gegenstände** $1,\ldots,N$**)** soll genau diese Idee verwirklichen, wobei K die Größe jeder Kiste, f_j den noch freien Platz in Kiste j und w_i die Größe von Gegenstand i angibt. Sie können davon ausgehen, dass $w_i \leq K$, $\forall i = 1,\ldots,N$, und $N > 0$ gilt.

Best-Fit(Gegenstände $1,\ldots,N$**)**
Eingabe: Gegenstände $1,\ldots,N$
Ausgabe: m Kisten, die die Gegenstände $1,\ldots,N$ beinhalten

```
m = 1; f_m = K;
für i = 1,...,N {
   bestbin = 0; bestplatz = ∞; gefunden = false;
   für j = 1,...,m {
      restplatz = f_j − w_i;
      falls restplatz ≥ 0 ∧ restplatz < bestplatz dann {
         bestplatz = restplatz;
         bestbin = j;
         gefunden = true;
      }
   }
   falls gefunden dann {
      packe Gegenstand i in Kiste bestbin;
      f_bestbin = f_bestbin − w_i;
   } sonst {
      m = m + 1;
      packe Gegenstand i in Kiste m;
      f_m = K − w_i;
   }
}
```

a) (8 Punkte) Führen Sie den Algorithmus **Best-Fit** für $K = 10$ auf die folgende Eingabereihenfolge aus und geben Sie an, welche Kiste welchen Gegenstand am Ende enthält:

Gegenstand i	1	2	3	4	5	6	7
Größe w_i	6	7	8	2	4	2	1

b) (4 Punkte) Welche (asymptotisch relative) Approximationsgüte ϵ gilt für **Best-Fit** jedenfalls? Begründen Sie Ihre Antwort.

c) (8 Punkte) Zeigen Sie mit einem Beispiel (mit mindestens 6 Gegenständen), dass dieses Verfahren nicht immer eine optimale Lösung liefert. Was können Sie aus *Ihrem Beispiel* über die Approximationsgüte schließen?

Aufgabe 2.B: Skiplisten **(18 Punkte)**

a) (6 Punkte)

Fügen Sie in eine anfangs leere randomisierte Skipliste S die folgenden Elemente gemäß ihrer zugehörigen Höhe in der vorgegebenen Reihenfolge ein:

Schlüssel	33	25	41	49	9	8	18	4	14
Höhe	1	0	1	0	0	3	1	0	2

Zeichnen Sie die resultierende Skipliste (nur das Endresultat).

b) (4 Punkte)

Suchen Sie nach der Zahl 27. Wieviele Schlüsselvergleiche waren hierfür notwendig? (Vergleiche mit ∞ zählen mit.)

c) (4 Punkte)

Suchen Sie nach der Zahl 41. Wieviele Schlüsselvergleiche waren hierfür notwendig? (Vergleiche mit ∞ zählen mit.)

d) (4 Punkte)

Es wird ein neues Element in eine randomisierte Skipliste eingefügt, wobei die Münzwurf-Methode aus der Vorlesung bzw. aus dem Skriptum verwendet wird. Wie groß ist die Wahrscheinlichkeit, dass der Container des neuen Elements die Höhe 5 annehmen wird?

Aufgabe 3.B: Theorie **(12 Punkte)**

a) Gegeben sei ein dynamisches Array A, in dem Elemente eingefügt werden können. Wenn A noch nicht voll ist, wird ein neues Element an eine beliebige freie Position eingefügt. Wenn A voll ist, wird seine Größe verdoppelt, um das neue Element aufzunehmen. Dabei kann es zu Umplatzierungen im Speicher kommen, was einen entsprechenden Aufwand verursacht.

- (2 Punkte)

  Geben Sie die **Worst-Case Laufzeit** für eine Einfüge-Operation in O-Notation an.

- (4 Punkte)

  Nehmen Sie an, Sie würden die Potenzialmethode für eine amortisierte Analyse auf dieses Problem anwenden. Welche Kenngröße kann für die Potenzialberechnung herangezogen werden?

- (3 Punkte)

  Geben Sie die **amortisierte Laufzeit** für eine Sequenz von n aufeinanderfolgenden Einfüge-Operationen in O-Notation an.

b) (3 Punkte)

Geben Sie in O-Notation an, wie groß die Höhe eines Fibonacci-Heaps maximal werden kann, der n Elemente enthält.