RWTH Aachen **SS 2018**
Lehrgebiet Theoretische Informatik Blatt 6
Rossmanith–Dreier–Kuinke 4.6.2018

Übung zur Vorlesung Algorithmen und Datenstrukturen

Aufgabe T12

Wir verwenden folgende universelle Familie von Hashfunktionen:

$$\{\, h_{a,b} \mid 1 \leq a < 5,\ 0 \leq b < 5 \,\}$$

mit $h_{a,b}(x) = ((ax + b) \bmod 5) \bmod 4$.
Berechnen Sie die Wahrscheinlichkeit, daß $h(2) = h(3)$ ist, falls wir h zufällig aus obiger Familie von Funktionen wählen.
Was müßte herauskommen, wenn man bedenkt, daß es sich um eine universelle Familie von Hashfunktionen handelt?
Da es sich um viele Funktionen handelt, sollte die Arbeit unter viele Personen verteilt werden, damit es schneller geht. Wie lange brauchen Sie gemeinsam, um diese Wahrscheinlichkeit ohne Taschenrechner oder ähnlichem zu berechnen?

Lösungsvorschlag:

Es gibt insgesamt vier Möglichkeiten für a und fünf Möglichkeiten für b, so daß wir 20 verschiedene $h_{a,b}$ in unserer Familie finden. Für jede dieser Funktionen müssen wir nun herausfinden, ob $h_{a,b}(2) = h_{a,b}(3)$ gilt. Wir berechnen also einfache beide benötigte Funktionswerte für alle 20 Funktionen:

a	b	$h_{a,b}(2)$	$h_{a,b}(3)$
1	0	2	3
1	1	3	0
1	2	0	0
1	3	0	1
1	4	1	2
2	0	0	1
2	1	0	2
2	2	1	3
2	3	2	0
2	4	3	0
3	0	1	0
3	1	2	0
3	2	3	1
3	3	0	2
3	4	0	3
4	0	3	2
4	1	0	3
4	2	0	0
4	3	1	0
4	4	2	1

Wir sehen, daß es genau zwei Kollisionen gibt. Da es 20 verschiedene Funktionen sind, ist die Wahrscheinlichkeit einer Kollision also genau 1/10. Aus der Theorie der universellen Hashfunktionen erwarten wir, daß die Kollisionswahrscheinlichkeit höchstens $1/m = 1/4$ beträgt. Mit 1/10 wird dieser garantierte Wert sogar noch unterboten.

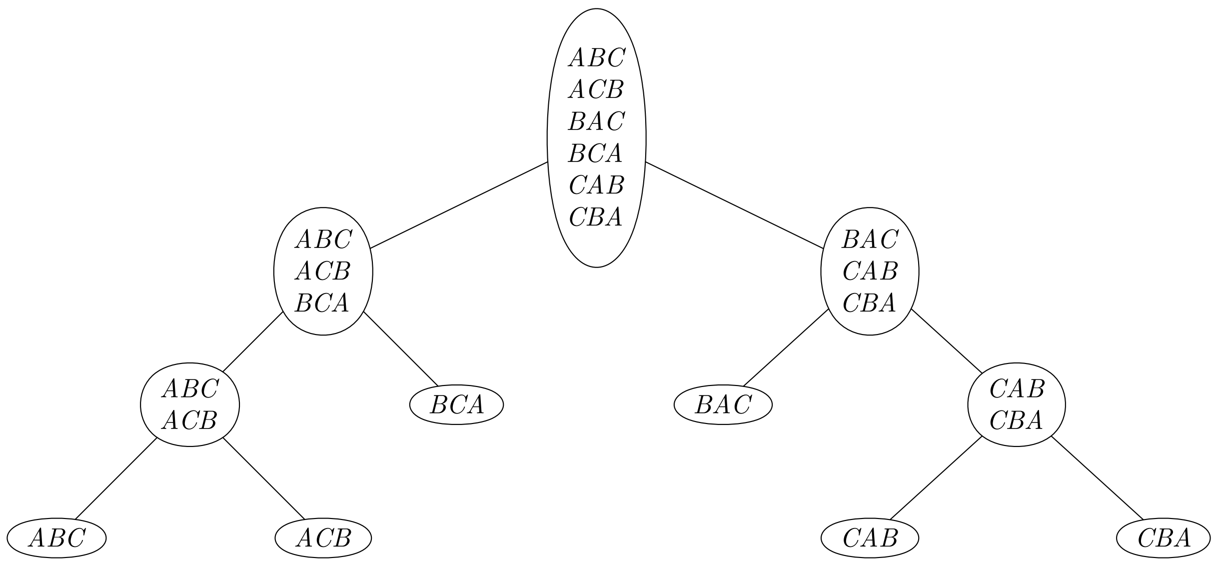

Abbildung 1: Vergleichsbaum für Aufgabe T13.

Aufgabe T13

Wir betrachten die drei Schlüssel A, B und C, welche wir in einem Array x der Größe drei in einer unbekannten Reihenfolge vorfinden.

Ein Algorithmus versucht herauszufinden, welche der möglichen sechs Permutationen vorliegt indem er zunächst den Vergleich $x[0] \leq x[1]$, dann den Vergleich $x[0] \leq x[2]$ und schließlich – falls noch nötig – den Vergleich $x[1] \leq x[2]$ durchführt.

Ist es möglich auf diese Weise die richtige Permutation zu finden? Wie sieht der entsprechende Vergleichsbaum aus? Wieviele Vergleiche würde ein auf dieser Methode basierender Sortieralgorithmus im Durchschnitt verwenden?

Lösungsvorschlag:

Die einzelnen Vergleiche partitionieren die Menge der möglichen Partitionen wie in Abbildung 1 dargestellt.

Man sieht, daß tatsächlich ein Vergleichsbaum entsteht, der nur einzelne Permutationen für jedes Blatt übrigläßt. Daher ist es in der Tat möglich, mit diesen Vergleichen die richtige Permutation zu bestimmen.

Wenn wir annehmen, daß jede Permutation mit gleicher Wahrscheinlichkeit, das heißt mit einer Wahrscheinlichkeit von 1/6, vorkommt, dann ist die erwartete Anzahl von Vergleichen $4 \cdot \frac{3}{6} + 2 \cdot \frac{2}{6} = \frac{8}{3}$.

Aufgabe H10 (10 Punkte)

Erstellen Sie mithilfe eines Programms eine Tabelle, welche die Wahrscheinlichkeiten von $h(x) = h(y)$ für alle $0 \leq x, y < 5$ enthält, falls h wieder zufällig aus der Familie von T12 gezogen wird.

Ist das Ergebnis das, was Sie erwarten?

Wiederholen Sie das Experiment, aber ersetzen Sie jetzt überall 5 durch 6. Kommentieren Sie das Ergebnis. Nehmen Sie insbesondere dazu Stellung, ob es sich auch jetzt um eine universelle Familie von Hashfunktionen handelt.

Lösungsvorschlag:
Das Programm in Abbildung 2 erledigt die Arbeit für $p = 5$ und $p = 6$, wobei es nicht die Wahrscheinlichkeiten, sondern die Gesamtzahl der Kollisionen ausgibt. Um Wahrscheinlichkeiten zu erhalten, muß man noch durch $p(p - 1)$ teilen, was aber häßliche Brüche ergibt.
Die Ausgabe des Programms ist:

```
H10a
20 2 2 2 2
2 20 2 2 2
2 2 20 2 2
2 2 2 20 2
2 2 2 2 20
H10b
30 4 14 12 14 4
4 30 4 14 12 14
14 4 30 4 14 12
12 14 4 30 4 14
14 12 14 4 30 4
4 14 12 14 4 30
```

Natürlich sind die Kollisionswahrscheinlichkeiten stets 1, wenn $x = y$. Ansonsten sind sie für $p = 5$ stets $1/10$. Wir erwarten natürlich, daß sie höchstens $1/4$ sind. Für $p = 6$ sieht es bitterer aus, da hier Kollisionswahrscheinlichkeiten von bis zu $7/15$ auftreten, was deutlich höher als $1/m = 1/4$ ist. Allerdings ist das Theorem aus der Vorlesung dadurch nicht widerlegt, weil ja 6 keine Primzahl ist. Ganz im Gegenteil: Dieses Beispiel demonstriert, wie wichtig es ist, eine Primzahl zu wählen.

Aufgabe H11 (4 Punkte)

Beweisen Sie, daß es keinen vergleichsbasierten Sortieralgorithmus gibt, welcher ein beliebiges Array der Größe fünf mit nur sechs Vergleichen sortieren kann.

Lösungsvorschlag:
In der Vorlesung wurde bewiesen, daß wir mindestens $\log(5!) = \log(120) \approx 6.9$ Vergleiche brauchen.

```
int p;
int h(int a, int b, int x) {
  return((a * x + b)%p)%4;
}
int countcollisions(int x, int y) {
  int a, b, count = 0;
  for(a = 1;  a < p;  a++) {
    for(b = 0;  b < p;  b++) {
      if(h(a, b, x) == h(a, b, y)) {
        count++;
      }
    }
  }
  return count;
}
void printtable() {
  int x, y;
  for(x = 0;  x < p;  x++) {
    for(y = 0;  y < p;  y++) {
      printf("%d ", countcollisions(x, y));
    }
    printf("\n");
  }
}
int main() {
  p = 5;
  printf("H7a\n");
  printtable();
  p = 6;
  printf("H7b\n");
  printtable();
}
```

Abbildung 2: Programm für H10.