Algorithmik Blatt 3 Teil 2

Mtr.-Nr. 6329857

Universität Hamburg — 7. November 2019

Aufgabe 8

```
1: procedure FindPairs(A)
2:     for i = 1 to A.length do
3:         for j = i+1 to A.length do
4:             if A[i] > A[j] then          // Ereignis A_{i,j}
5:                 output A[i], A[j]
```

$A_{i,j}$ sei das Ereignis, dass die Bedingung A in Iteration (i, j) erfüllt ist. Wir nehmen an, dass die Ereignisse $A_{i,j}$ für alle i, j unabhängig voneinander sind und interessieren uns für die Vereinigung aller Ereignisse, also für das Gesamtereignis, dass mindestens eines der Teilereignisse eintritt, wobei n die Länge des Array ist:

$$\begin{aligned} A_{i,j} &= A[i] > A[j] \\ A(n) &= \bigcup_{i=1}^{n} \bigcup_{j=i+1}^{n} A_{i,j} \end{aligned} \tag{1}$$

Wir definieren die Indikatorfunktion für $A(n)$:

$$I\{A(n)\} = \begin{cases} 1 & \text{wenn } \bigcup_{i=1}^{n} \bigcup_{j=i+1}^{n} A_{i,j} \\ 0 & \text{sonst} \end{cases} \tag{2}$$

Und bestimmen nun den zugehörigen Erwartungswert:

$$\begin{aligned} E[A(n)] &= E\left[\bigcup_{i=1}^{n} \bigcup_{j=i+1}^{n} A_{i,j}\right] \\ &= I\left\{\bigcup_{i=1}^{n} \bigcup_{j=i+1}^{n} A_{i,j}\right\} \cdot \Pr\left(\bigcup_{i=1}^{n} \bigcup_{j=i+1}^{n} A_{i,j}\right) \\ &= 1 \cdot \sum_{i=1}^{n} \sum_{j=i+1}^{n} \Pr\left(A_{i,j}\right) \end{aligned} \tag{3}$$

Da die Werte im Array paarweise verschieden und zufällig verteilt sind und $i \neq j$, können wir für $\Pr(A_{i,j}) = \frac{1}{2}$ annehmen.

$$
\begin{aligned}
E[A(n)] &= \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{1}{2} \\
&= \sum_{i=1}^{n} \left(\sum_{j=1}^{n} \frac{1}{2} - \sum_{j=1}^{i} \frac{1}{2} \right) \\
&= \sum_{i=1}^{n} \left(\frac{1}{2} \cdot n - \frac{1}{2} \cdot i \right)
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
E[A(n)] &= \frac{1}{2} \cdot \sum_{i=1}^{n} (n-i) \\
&= \frac{1}{2} \cdot \sum_{i=1}^{n} (n-i) \\
&= \frac{1}{2} \cdot \left(\sum_{i=1}^{n} n - \sum_{i=1}^{n} i \right) \\
&= \frac{1}{2} \cdot \left(n^2 - \frac{n^2+n}{2} \right) \\
&= \frac{n^2-n}{4}
\end{aligned} \tag{5}
$$

Die Erwartete Anzahl an Ausgaben bei einem Array der Länge n ist $\frac{n^2-n}{4}$

Aufgabe 9

Es ist eine gdw-Aussage zu zeigen, also sind beide Richtungen getrennt zu zeigen.

Beweis Richtung 1: Pfad-Existenz ⇒ Knoten-Eigenschaft

Jeder Knoten, der als Zielknoten einer Kante im Pfad vorkommt, muss in der darauf folgenden Kante als Ursprungsknoten vorkommen. Der Zielknoten der letzten Kante muss als Ursprungsknoten der ersten Kante vorkommen. Daraus folgt, dass jeder Knoten genau so oft als Ursprungs wie als Zielknoten in den Kanten des Pfades vorkommen muss.

Zusammen mit der Forderung, dass der Pfad genau so viele Kanten wie der gesamte Graph enhält und keine Kanten doppelt enthalten darf, ergibt sich, dass er genau alle Kanten enthält.

Da also in der Menge aller Kanten jeder Knoten genau so oft als Ursprungs wie als Zielknoten vorkommt, muss für alle Knoten v gelten, dass $\text{outdeg}(v) = \text{indeg}(v)$.

Aus der Forderung, dass der Graph zusammenhängend ist, folgt dass für alle Knoten v $\deg(v) \geq 1$ ist, denn sobald ein Pfad existiert (der Graph ist weder leer noch besteht er aus nur einem kantenlosen Knoten), muss jeder Knoten erreichbar sein, also mindestens eine eingehende Kante besitzen.

Beweis Richtung 2: Pfad-Existenz $\Leftarrow$ Knoten-Eigenschaft

Wir erinnern uns, dass der Graph (a) streng zusammenhängen ist, (b) jeder Knoten genau so viele eingehende wie ausgehende Kanten hat, und (c) mindestens je eine ausgehende und eingehende Kante hat.

Unter diesen Annahmen können wir einen Kantendefinierten Pfad, der alle Kanten enthält wie folgt konstruieren:

1. Wähle einen beliebigen Startknoten s
2. Wähle eine ausgehende Kante (wir wissen wegen (c), dass eine existiert), gehe über sie zum Nachbarknoten und markiere sie als benutzt.
3. Gehe so von Knoten zu Knoten ohne eine bereits benutzte Kante erneut zu verwenden.

Wir wissen durch (b), dass wir jeden Knoten, den wir erreichen, auch wieder verlassen können. Wir wissen auch, dass wir den Startknoten wieder erreichen können, weil der Graph streng zusammenhängend ist und der Startknoten für jedes mal, dass wir ihn verlassen auch noch eine unbenutzte Eingangskante für die Rückkehr besitzt.

An jedem Knoten an dem wir die Wahl zwischen mehreren Ausgangskanten haben, wissen wir, dass wenn wir eine wählen, auch wieder zurückkehren können um später die andere zu wählen, weil es zu dieser auch eine Eingangskante geben muss, um (b) zu erfüllen.

Es kann passieren, dass wir die Kanten so wählen, dass wir zum Startknoten zurückgelangen ohne alle Kanten besucht zu haben und dabei auch schon alle seine Ausgangskanten verbraucht zu haben. Auf Grund von (a) wissen wir aber, dass die nicht besuchten Kanten prinzipiell vom Startknoten aus erreichbar gewesen wären. Es gibt also einen Knoten c an dem wir eine andere Kante hätten wählen können. Immer wenn dieser Fall eintritt, können wir den Vorgang mit c als neuem Startknoten wiederholen und finden so einen durch c laufenden Zyklus aus bisher unbesuchten Kanten, den wir dann in unseren zuvor unvollständigen durch s laufenden Zyklus einspleißen können. Das lässt sich so lange wiederholen bis alle Kanten genau einmal benutzt wurden.

Somit ist per Konstruktion gezeigt, dass für jeden Graph, der die geforderten Eigenschaften erfüllt, der gefragte Pfad konstruiert werden kann.