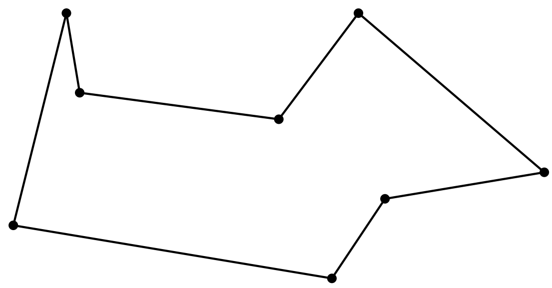
# Theoretische Informatik 1
# 1. Übungsblatt

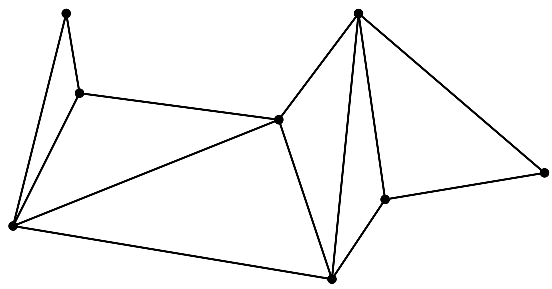
Adam Duck

24. Oktober 2001

**1.1** Ein einfaches Polygon ist ein von einem geschlossenen, sich nicht schneidenden, Streckenzug begrenztes, ebenes geometrisches Objekt. Hier ein Beispiel für ein einfaches Polygon $G$ mit $n = 8$ Ecken.

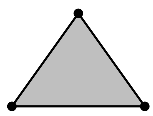


Wir möchten ein einfaches Polygon triangulieren, das heißt in disjunkte Dreiecke zerlegen. Dazu können nicht überschneidende Diagonalen in das einfache Polygon eingefügt werden, wobei eine Diagonale eine im Polygon verlaufende Strecke ist, die zwei Eckpunkte verbindet. Hier ist ein einfaches Beispiel einer möglichen Triangulation des obigen einfachen Polygons:



(a) **Beweise**, dass sich jedes einfache Polygon $G$ mit $n$ Ecken durch $n-3$ Diagonalen triangulieren lässt und, dass dabei $n-2$ Dreiecke entstehen. (7 Punkte)
**Lösung**:
Wir benutzen hier – wie vorgeschlagen – die vollständige Induktion.

**IV** für $n = 3$



$n - 3 = 0$ Diagonalen $\surd$
$n - 2 = 1$ Dreieck $\surd$

**IS** $n \to n+1$

Wir nehmen an, das Polygon $G$ mit $n$ Kanten sei schon trianguliert. Somit hat es $n-3$ Diagonalen und $n-2$ Dreiecke.

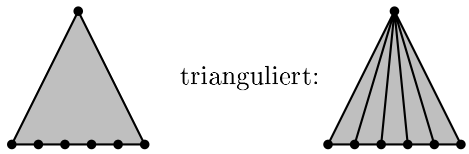
Wir fügen nun einen Punkt an beliebiger Stelle in das Polygon ein. Dies tun wir, indem wir eine Kante herausnehmen und den neuen Punkt mit den zwei Punkten verbinden, die im Polygon $G$ durch die herausgenommene Kante verbunden waren. Damit haben wir eine neues Polygon $G'$ mit 1 neuen Kante (# der Kanten im Polygon $G'$: $n+1$) erhalten. Wir haben ein Dreieck zerstört und ein Viereck erhalten.

Nun triangulieren wir dieses Viereck mit 1 neuen Diagonalen (# der Diagonalen in $G'$: $n-3+1=(n+1)-3$), womit wir dann 1 neues Dreieck erhalten (# der Dreiecke in $G'$: $n-2+1=(n+1)-2$).

(b) **Beschreibe** für allgemeine Eckenanzahl $n$ ein einfaches Polygon, bei dem diese Zerlegung eindeutig ist, d.h. nur durch eine einzige Menge von Diagonalen kann die Triangulation erfolgen. Aus Deiner Beschreibung soll erkennbar sein, wie man für gegebenes $n$ ein solches einfaches Polygon konstruiert. Argumentiere, warum die Zerlegung deiner einfachen Polygone eindeutig ist. (3 Punkte)

**Lösung**:

Das einfachste Polygon, das ich finden konnte, sieht aus wie ein Dreieck, wobei sich alle Punkte, bis auf einen, auf einer Geraden befinden:




In einem solchen Polygon können Linien, die Punkte des Polygons miteinander Verbinden, *nur* auf eine Weise gezeichnet werden: verbunden mit dem nach oben exponierten Punkt. Somit gibt es auch für die Diagonalen der Triangulation nur eine mögliche Lösung.

**1.2** Wir betrachten die Menge $U = \{1, 2, \dots, n\}$. In einem ersten Experiment bestimmen wir eine Menge $A \subseteq U$, indem wir jedes Element aus $U$ mit einer Wahrscheinlichkeit $p_A$ in $A$ aufnehmen. Wir wiederholen das Experiment und bilden eine Menge $B$, wobei wir jedes Element aus $U$ mit der Wahrscheinlichkeit $p_B$ in $B$ aufnehmen.

(a) **Bestimme** $E(|A \cap B|)$, den Erwartungswert der Mächtigkeit der Schnittmenge. (3 Punkte)

**Lösung**:

Beginnen wir mit der Menge $A$. Da jedes Element mit einer Wahrscheinlichkeit $p_A$ in $A$ hineinkommt und wir dieses $n$-mal machen, ist die $E(|A|) = n \cdot p_A$. Dasselbe gilt für $B$. Nun ist aber die Wahscheinlichkeit, dass ein Element sowohl in $A$ als auch in $B$ kommt $p_A \cdot p_B$, denn das ist praktisch ein zweistufiges Experiment. Damit bekommen wir

$$E(|A \cap B|) = n \cdot p_A p_B$$

(b) **Bestimme** $E(|A \cup B|)$, den Erwartungswert der Mächtigkeit der Vereinigungsmenge. (4 Punkte)

**Lösung**:

Hier ist es einfach die Formel die man von Stochastik her kennt:

$$\begin{aligned} E(|A \cup B|) &= E(|A|) + E(|B|) - E(|A \cap B|) \\ &= np_A + np_B - np_A p_B \\ &= n(p_A + p_B - p_A p_B) \end{aligned}$$

**1.3 Bestimme** exakt den Wert, den $m$ nach Aufruf der folgenden Programme jeweils in Abhängigkeit von $n > 0$ annimmt.

(a)
```
int procA(int n) {
    int m=1;
    int i=1;
    while(i < n) {
        m=m+i;
        i=i*2;
    }
    return m;
}
```

(3 Punkte)

**Lösung**:
Als erstes möchte ich mir anschauen, woraus $m$ besteht bzw. wie es aufgebaut wird: $m = 1 + 1 + 2 + 4 + 8 + 16 + \ldots$. Das Einzige, das hier Probleme bereitet, ist die Abbruchbedingung, denn $i$ wird jedesmal um eine 2-er Potenz größer. Das können wir aber lösen, indem wir eine 2. Variable einführen, die nur linear wächst und uns somit die Schleifendurchläufe zählt. Der betreffende Programmteil sähe dann folgendermaßen aus:

```
while(2^x < n) {
    m=m+2^x;
    x=x+1;
}
```

Damit können wir die Abbruchbedingung auflösen

Ich darf die rechte Seite auf keinen Fall kleiner machen.

$$\begin{aligned} 2^x &< n \\ x &< \lceil ld(n) \rceil \end{aligned}$$

und es ergibt sich für $m$

$$m = 1 + \sum_{k=0}^{\lceil ld(n) \rceil - 1} 2^k = 1 + \left(2^{\lceil ld(n) \rceil - 1 + 1} - 1\right) = 2^{\lceil ld(n) \rceil}$$

(b)
```
int procB(int n) {
    int i,j,k;
    int m=0;
    for(i=0;i<=n;i++)
        for(j=i;j<=2*i;j++)
            for(k=1;k<=j;k++)
                m++;
```

```
    return m;
}
```

(4 Punkte)

**Lösung**:
Hier lassen sich die `for`-Schleifen – wie in der Vorlesung – sehr gut als Summen ineinander schreiben. Da jedes Mal nur eine eins hinzuaddiert wird, sieht die Berechnung für $m$ folgendermaßen aus:

$$\begin{aligned}
m &= \sum_{i=0}^{n} \sum_{j=i}^{2i} \sum_{k=1}^{j} 1 = \sum_{i=0}^{n} \sum_{j=i}^{2i} j \\
&= \sum_{i=0}^{n} \left( \sum_{j=i}^{2i} j + \sum_{j=1}^{i-1} j - \sum_{j=1}^{i-1} j \right) \\
&= \sum_{i=0}^{n} \left( \frac{2i(2i+1)}{2} - \frac{(i-1)i}{2} \right) = \frac{1}{2} \sum_{i=0}^{n} \left(3i^2 + 3i\right) \\
&= \frac{3}{2} \left( \frac{1}{6}(n(n+1)(2n+1) + \frac{1}{2}n(n+1) \right) \\
&= \frac{3}{2} \left( \frac{1}{6}(n^2+n)(2n+1) + \frac{3n^2+3n}{6} \right) \\
&= \frac{1}{4} \left(2n^3 + n^2 + 2n^2 + n + 3n^2 + 3n\right) \\
&= \frac{1}{4} \left(2n^3 + 6n^2 + 4n\right) = \frac{1}{2} \left(n^3 + 3n^2 + 2n\right) = \frac{1}{2} n \left(n^2 + 3n + 2\right) \\
m &= \frac{n(n+1)(n+2)}{2}
\end{aligned}$$