Klausur

18. Februar 2005, 10:15 - 12:15 Uhr

Name: ....

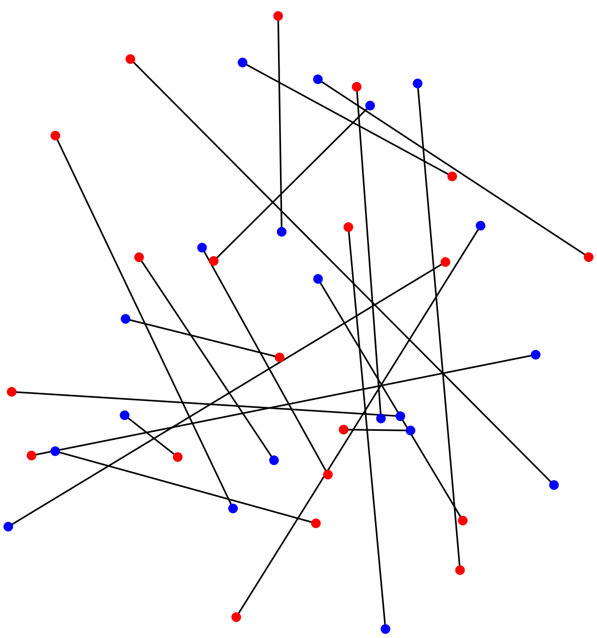
Matrikelnummer: ....

Anzahl beschriebener Blätter
(ohne Aufgabenblatt): ....

Bitte beschriften Sie jedes weitere Blatt mit Name und Matrikelnummer!

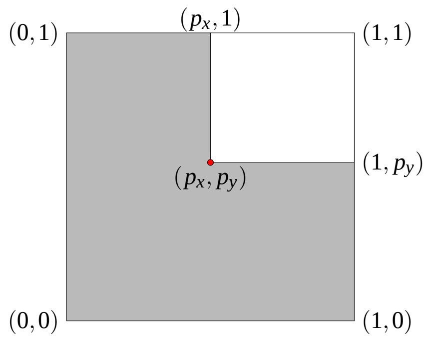
Bearbeitungszeit:	120 Minuten	Gesamtzahl Aufgaben:	5
Zugelassene Hilfsmittel:	Keine!	Gesamtpunktzahl:	50

Aufgabe 1 (12 PUNKTE)
Sei S eine Menge von n Strecken in der Ebene. Geben Sie einen Algorithmus an, der in Zeit $O(n \log n)$ (im Einheitskostenmaß) testet, ob die Strecken in S paarweise disjunkt sind. Sie dürfen annehmen, dass sich die Strecken in „allgemeiner Lage“ befinden, d.h., alle Streckenendpunkte haben paarweise verschiedene x-Koordinaten. Insbesondere gibt es also keine vertikalen Strecken. Begründen Sie die Laufzeit und die Korrektheit ihres Algorithmus.



Aufgabe 2 (9 PUNKTE)

Sei S eine Menge von n Punkten in der Ebene. Jeder Punkt $p = (p_x, p_y)$ aus S habe Koordinaten zwischen Null und Eins, d. h. $0 < p_x < 1$ und $0 < p_y < 1$. Für jeden Punkt $p \in S$ sei $L(p)$ das Polygon mit Ecken $(0,0), (1,0), (1, p_y), (p_x, p_y), (p_x, 1), (0, 1)$.



Sei $S' \subseteq S$ die Menge aller Punkte in S, für die jeder Punkt aus S in dem Polygon $L(p)$ oder auf dem Rand von $L(p)$ liegt. Geben Sie einen Algorithmus an, der in Zeit $O(n \log n)$ den (bzw. einen) Punkt aus S' berechnet, für den die Fläche des Polygons $L(p)$ minimal ist. Begründen Sie die Laufzeit ihres Algorithmus. Zur Korrektheit brauchen Sie ausnahmsweise nichts zu sagen.

Aufgabe 3 (12 PUNKTE)

(a) Sei $\Sigma = \{0, 1, \$\}$. Geben Sie einen regulären Ausdruck für die Sprache $L = \{w \in \Sigma^* \mid w \text{ beginnt mit 0 und endet mit } \$\}$ an.

(b) Seien A und B Sprachen und sei $B \in \mathbf{NP}$. Falls $A \preceq_{\mathbf{P}} B$, liegt dann auch A in $\mathbf{NP}$? Begründen Sie ihre Antwort.

(c) Gibt es Sprachen über dem Alphabet $\Sigma = \{0, 1, \$\}$, die nicht in **EXP** liegen? Begründen Sie ihre Antwort.

(d) Geben Sie die formale Definition von **NP**-Vollständigkeit an.

(e) Wie kann man von einer Sprache beweisen, dass sie **NP**-vollständig ist, ohne die Definition zu benutzen? Erklären Sie, warum dies richtig ist.

Aufgabe 4 (9 PUNKTE)

Sei $\Sigma = \{\$, 0, 1\}$ und seien

$$\begin{aligned} SOS \quad := \quad & \{\$\mathrm{bin}(a_1)\$\$\mathrm{bin}(a_2)\$ \cdots \$\mathrm{bin}(a_n)\$\$\mathrm{bin}(b)\$ \in \Sigma^* \mid \\ & \text{es gibt Indexmenge } I \subseteq \{1, \ldots, n\} \text{ so dass } \sum_{i \in I} a_i = b\} \end{aligned}$$

und

$$\begin{aligned} RS \quad := \quad & \{\$\mathrm{bin}(g_1)\$\$\mathrm{bin}(g_2)\$ \cdots \$\mathrm{bin}(g_m)\$\$\mathrm{bin}(k_1)\$\$\mathrm{bin}(k_2)\$ \cdots \$\mathrm{bin}(k_m)\$\$\mathrm{bin}(G)\$\$\mathrm{bin}(K)\$ \in \Sigma^* \mid \\ & \text{es gibt } c_1, \ldots, c_m \in \{0, 1\} \text{ so dass } \sum_{i=1}^{m} c_i g_i \leq G \text{ und } \sum_{i=1}^{m} c_i k_i \geq K\} \end{aligned}$$

Zeigen Sie, dass $SOS \preceq_{\mathbf{P}} RS$ gilt.

Aufgabe 5 (8 PUNKTE)

(a) Zeigen Sie, dass $4n^2 + 8n = O(n^2)$.

(b) Im Folgenden sei $A[1..n]$ ein Feld von Punkten in der Ebene. Bestimmen Sie jeweils eine möglichst gute asymptotische obere Schranke für die Laufzeit im schlechtesten Fall (im Einheitskostenmaß) für die folgenden beiden ineffizienten(!) Algorithmen:

```
Algorithmus hull_edges(A, n)
for i := 1 to n
do for j := 1 to n
   do if (i ≠ j)
      then ok := true;
           for k := 1 to n
           do if (Punkt A[k] liegt rechts von oder auf der Strecke A[i]A[j] )
              then ok := false;
              endif
           endfor
           if (ok)
           then gib die Strecke A[i]A[j] aus;
           endif
      endif
   endfor
endfor
```

```
Algorithmus weak_extreme_points(A, n)
for i := 1 to n
do for j := 1 to n
   do for k := 1 to n
      do for l := 1 to n
         do if ( l ≠ i und l ≠ j und l ≠ k )
            then if (Punkt A[l] liegt im Dreieck Δ(A[i], A[j], A[k]))
                 then markiere A[l];
                 endif
            endif
         endfor
      endfor
   endfor
endfor
for i := 1 to n
do if (A[i] ist nicht markiert)
   gib A[i] aus;
endfor
```