

Projektaufgabe 2

WPF-Modul *Cluster Computing*

Zusammenhängende Komponenten

Ausgangspunkt ist ein matrixförmiger Pixelbereich P mit m Zeilen und n Spalten. Jedes Pixel p des Bereichs soll genau eine der Eigenschaften *schwarz* ($c(p) = 1$) oder *weiß* ($c(p) = 0$) aufweisen und über eine Zeilenkoordinate i mit $0 \leq i \leq m - 1$ und eine Spaltenkoordinate j mit $0 \leq j \leq n - 1$ identifizierbar sein: $p(i,j)$.

Für $p(i,j) \in P$ wird eine Nachbarschaft $N(p)$ wie folgt definiert:

$$N(p(i,j)) = \{ p(k,l) : |k - i| \leq 1 \wedge |l - j| \leq 1 \wedge 0 \leq k \leq m - 1 \wedge 0 \leq l \leq n - 1 \} \setminus \{ p(i,j) \}$$

Damit werden für jedes nicht an einem Rand von P liegende Pixel 8 Nachbarn bestimmt.

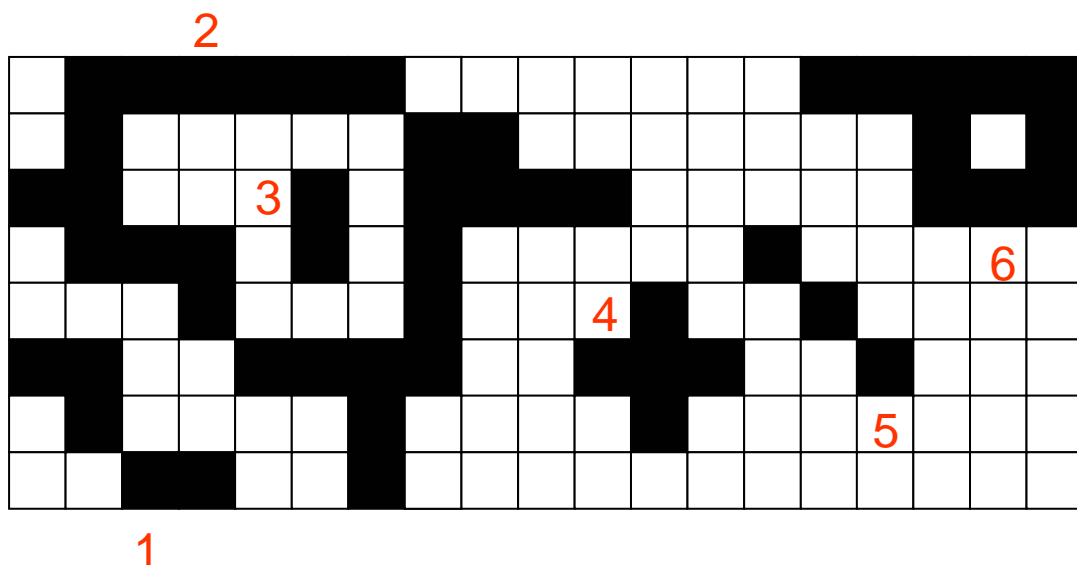
Zusammenhängende schwarze Komponenten

Eine nichtleere Pixelmenge $Z \subseteq P$ heißt zusammenhängende schwarze Komponente genau dann, wenn folgende Eigenschaften erfüllt sind:

- Für alle $p \in Z$ gilt $c(p) = 1$
- Für beliebige $p, p' \in Z$ mit $p \neq p'$ gilt:
Es gibt eine Folge (p_1, \dots, p_r) von Pixeln aus Z mit $p_1 = p, p_r = p'$ und $p_{i+1} \in N(p_i)$ für $1 \leq i \leq r - 1$

Maximale zusammenhängende schwarze Komponenten

Eine zusammenhängende schwarze Komponente Z heißt maximal genau dann, wenn es kein Pixel $p \in Z$ gibt, in dessen Nachbarschaft $N(p)$ sich ein schwarzes Pixel $q \in P \setminus Z$ befindet.



Die Abbildung zeigt 6 maximale zusammenhängende schwarze Komponenten mit jeweils 5, 27, 2, 5, 3 und 10 Pixeln (in der Reihenfolge der Komponentenummerierung).

Aufgabenstellung:

Entwickeln Sie einen parallelen Ansatz zur Identifizierung aller maximalen zusammenhängenden schwarzen Komponenten eines Pixelbereichs P und setzen Sie diesen Ansatz in einem C-Programm unter Einsatz von MPI um. Als Resultat des Programms soll

- die Anzahl dieser Komponenten
- die Anzahl der Pixel je Komponente und
- ein Pixel als Repräsentant für jede Komponente

ausgegeben werden. Schreiben Sie einen Generator, der Test-Pixelbereiche erzeugt. Sehen Sie Möglichkeiten zur Generierung von Pixelmustern vor, die auch für große Pixelbereiche eine einfache Überprüfung der Resultate Ihres Programms gestatten.

Analysieren Sie das Laufzeitverhalten Ihres Programmes!