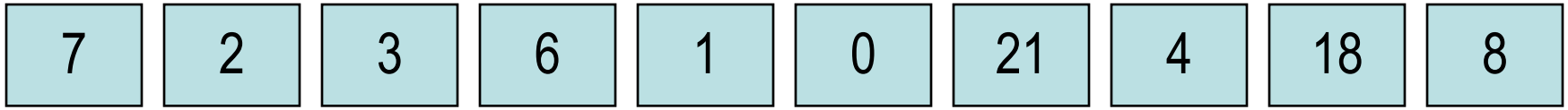
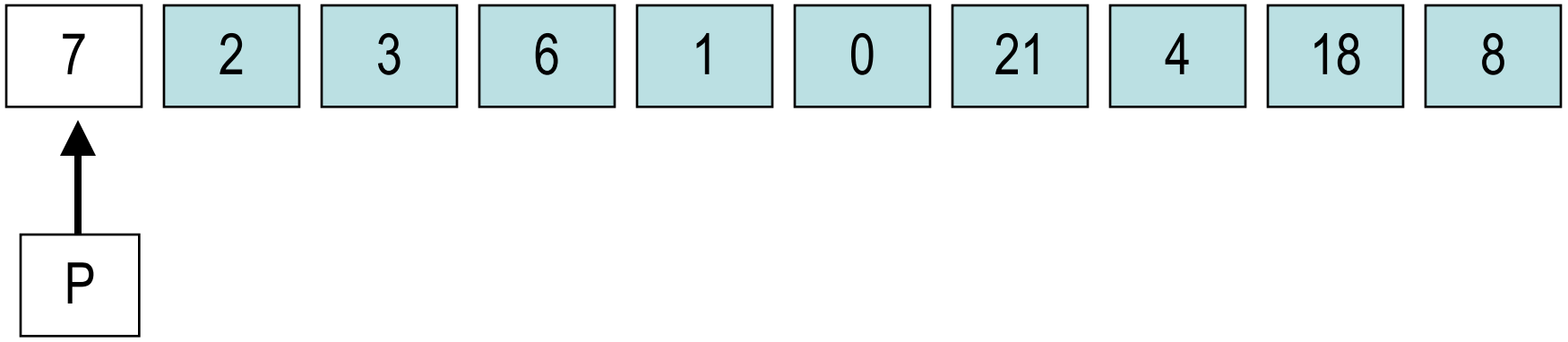
Funktionsweise D0



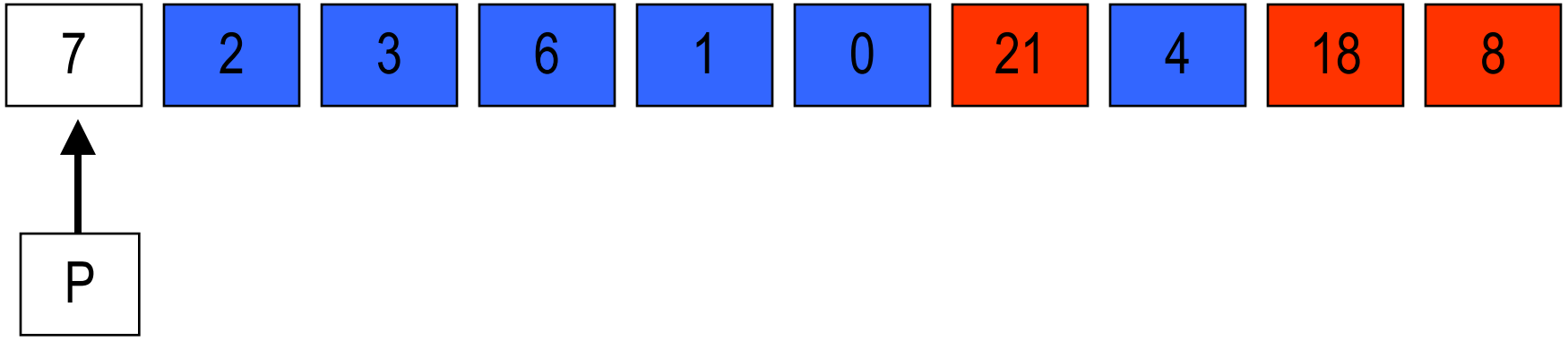
Wir wollen nun das obige Array sortieren

Funktionsweise D1



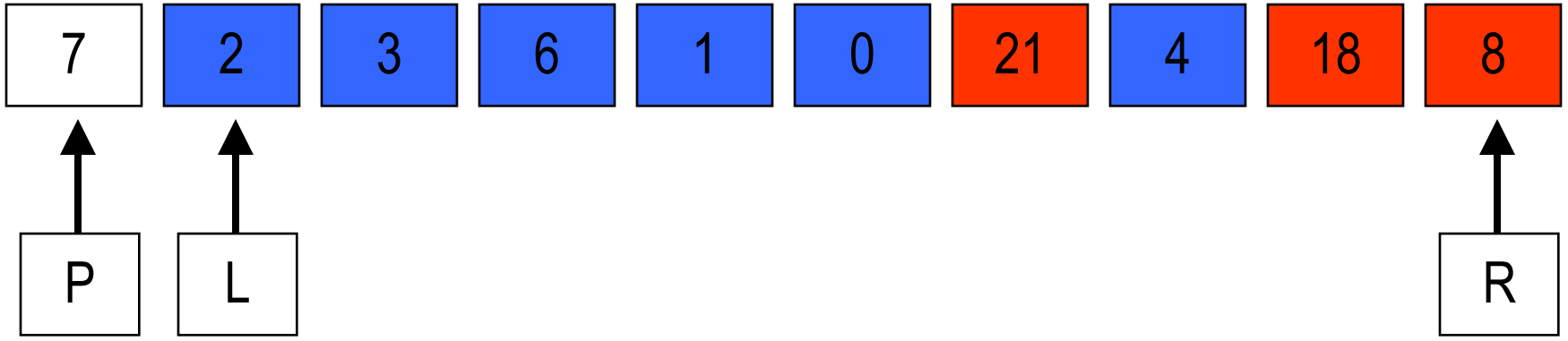
Zuerst wird das Pivotelement ausgewählt. Dies kann auf verschiedene Arten erfolgen. Wir entscheiden uns hier für das am weitesten links stehende Element.

Funktionsweise D2



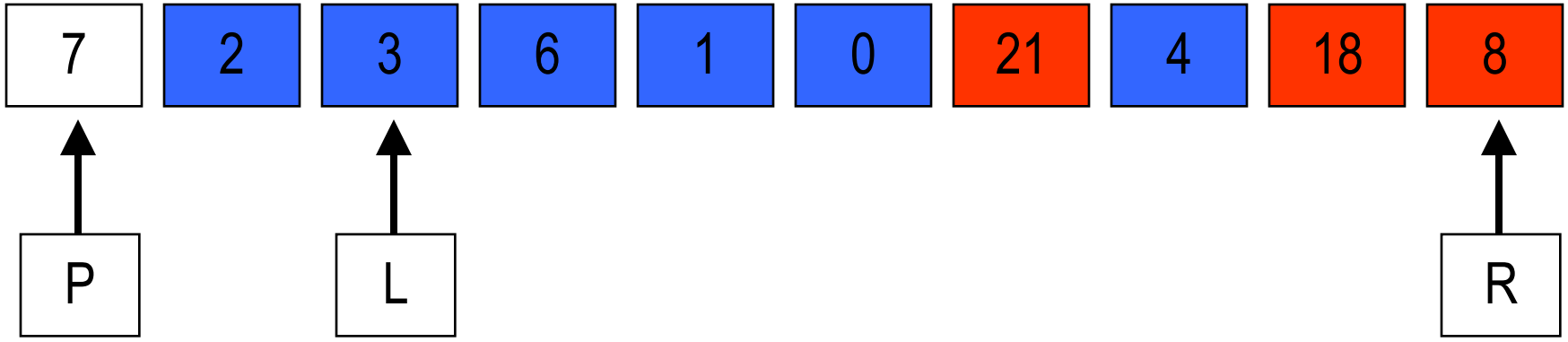
Zur besseren Übersicht färben wir die Elemente die größer als das Pivotelement sind rot und die kleineren blau ein.

Funktionsweise D3

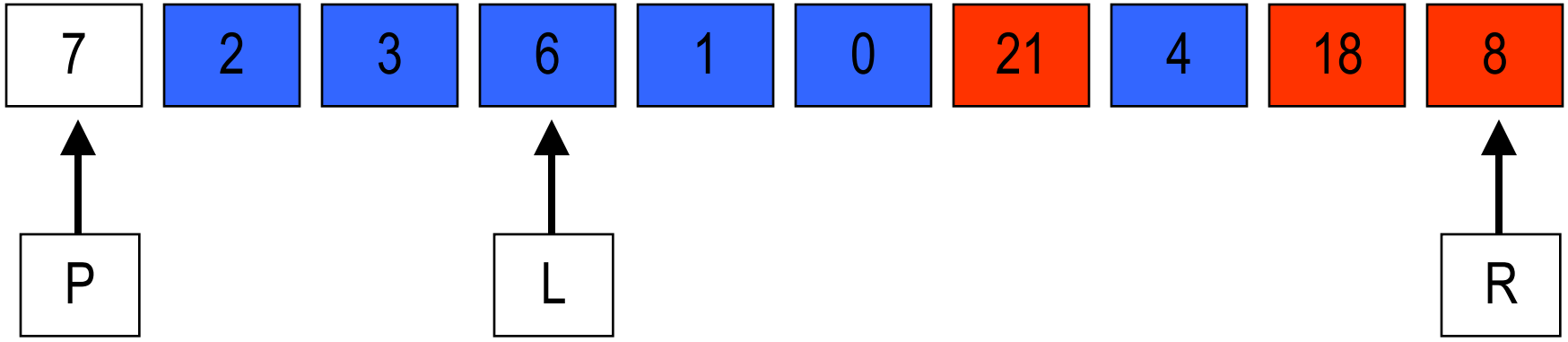


Nun werden die Pointer gesetzt. Wir lassen nun den linken Pointer so lange nach rechts wandern, bis er auf ein Element zeigt, das größer als das Pivotelement oder gleich groß ist (also ein „rotes Element“).

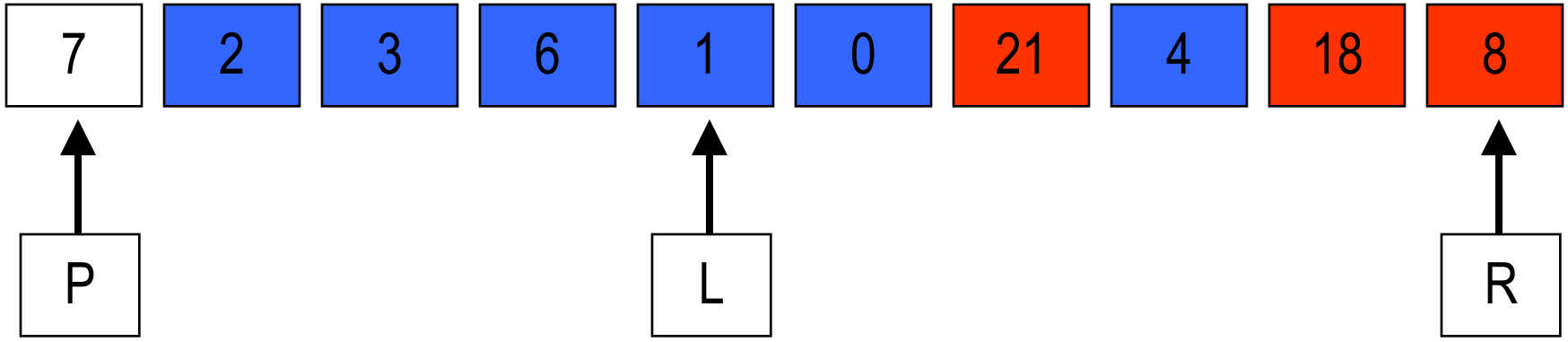
Funktionsweise D4



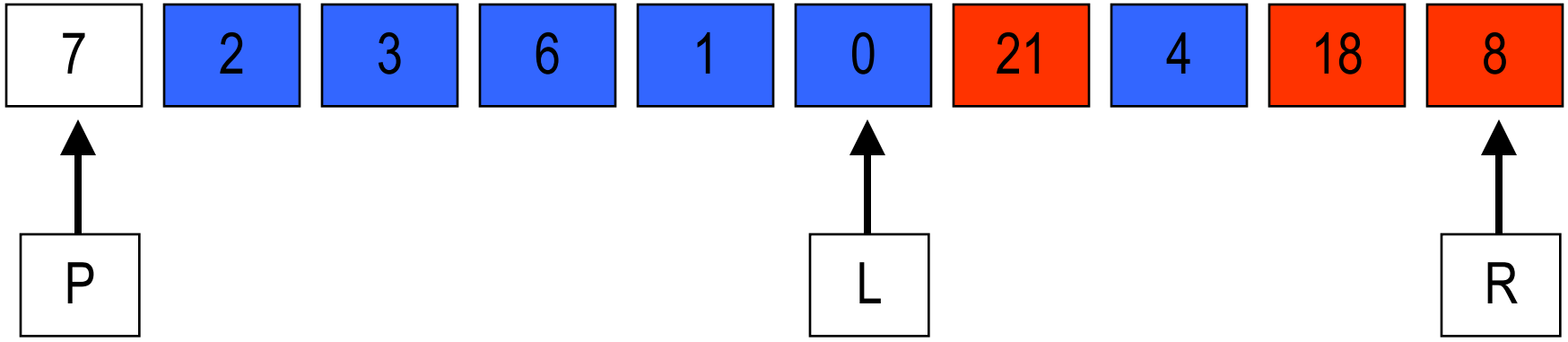
Funktionsweise D5



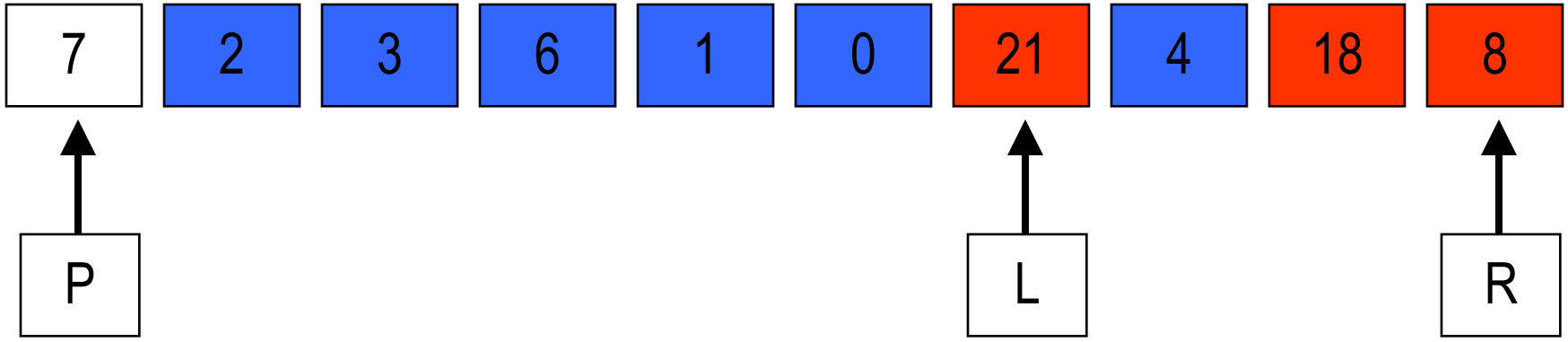
Funktionsweise D6



Funktionsweise D7

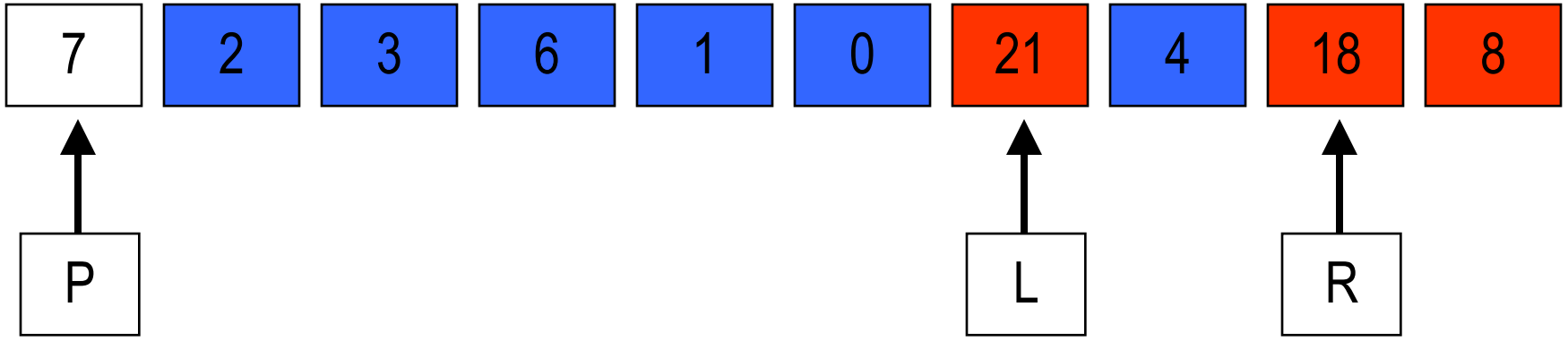


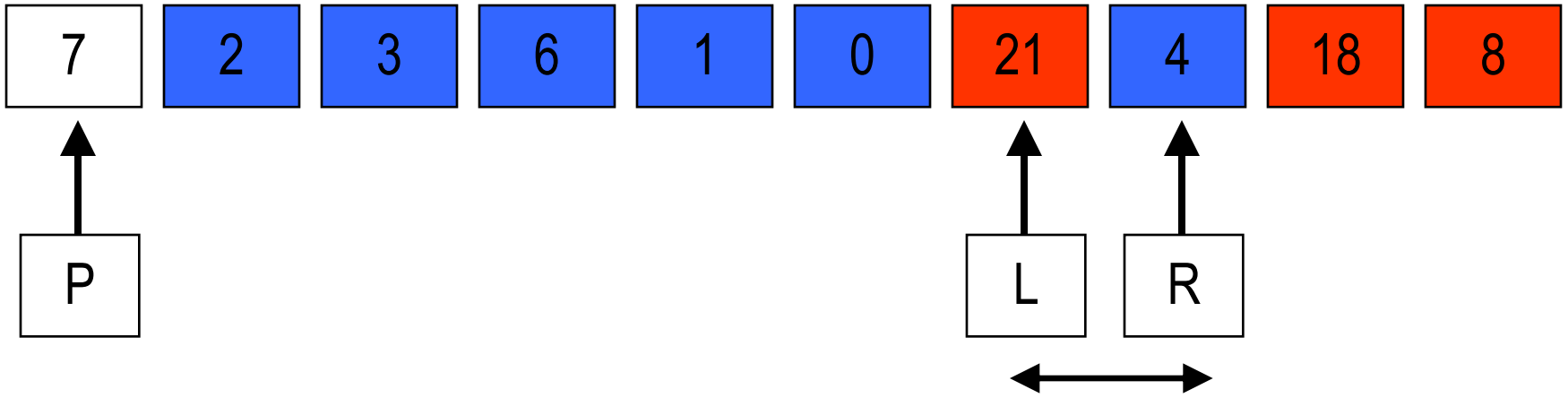
Funktionsweise D8



Der linke Pointer zeigt nun auf ein größeres Element. Nun lassen wir den rechten Pointer nach links wandern, bis er auf ein Element zeigt, das kleiner als das Pivotelement oder gleich groß ist.

Funktionsweise D9

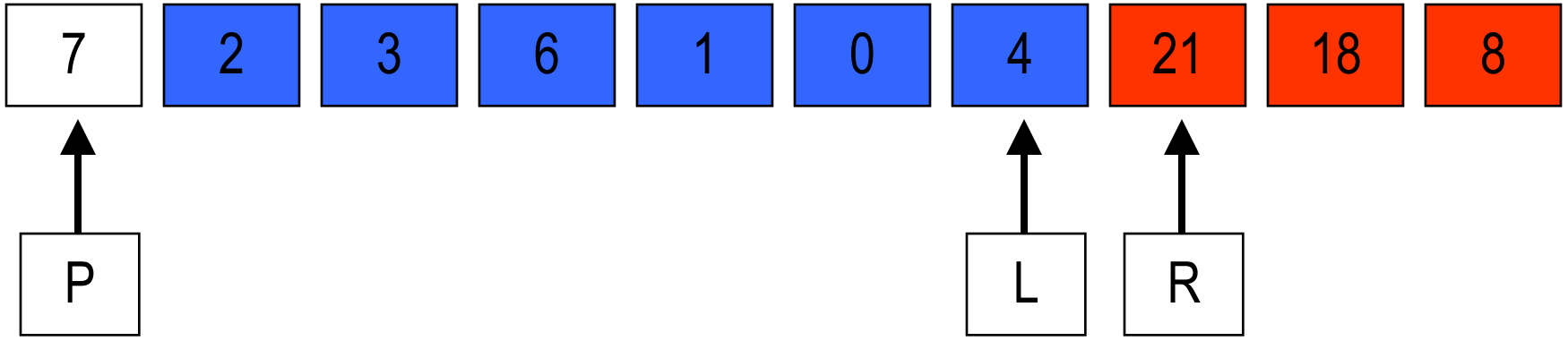




Der rechte Pointer zeigt nun auf ein Element, das kleiner als das Pivotelement ist.

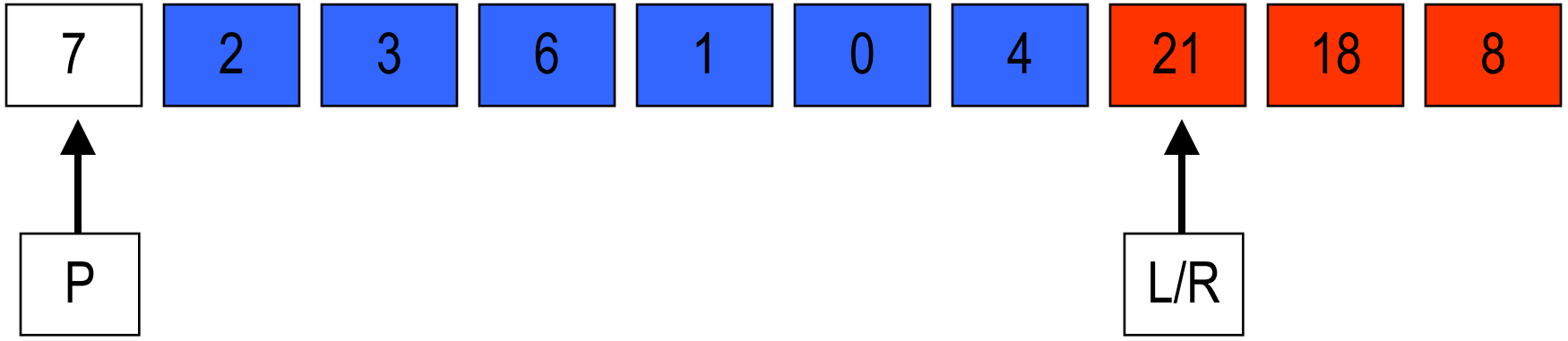
Wir vertauschen nun die beiden Elemente.

Funktionsweise D11



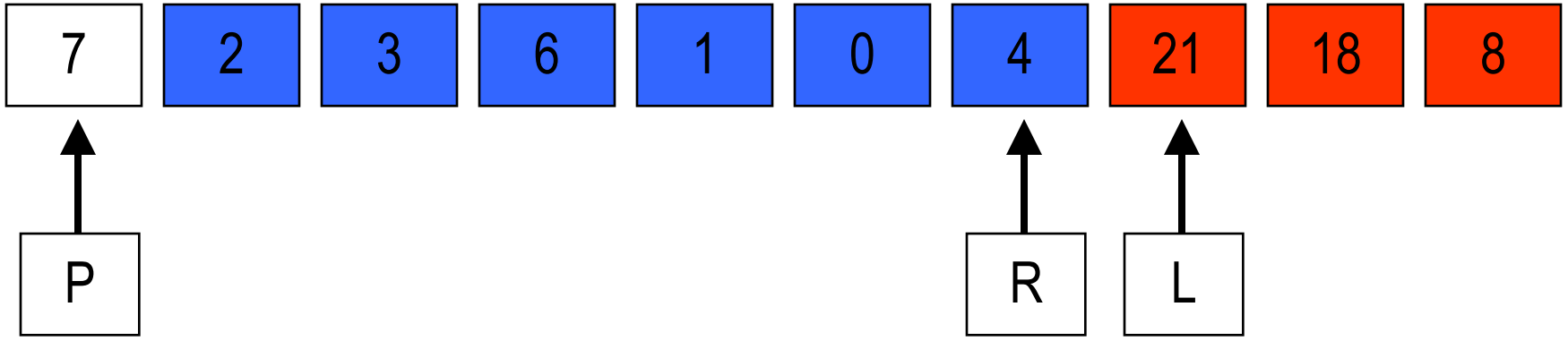
Wir lassen den linken Pointer bis zum nächsten „roten Wert“ weiterwandern.

Funktionsweise D12

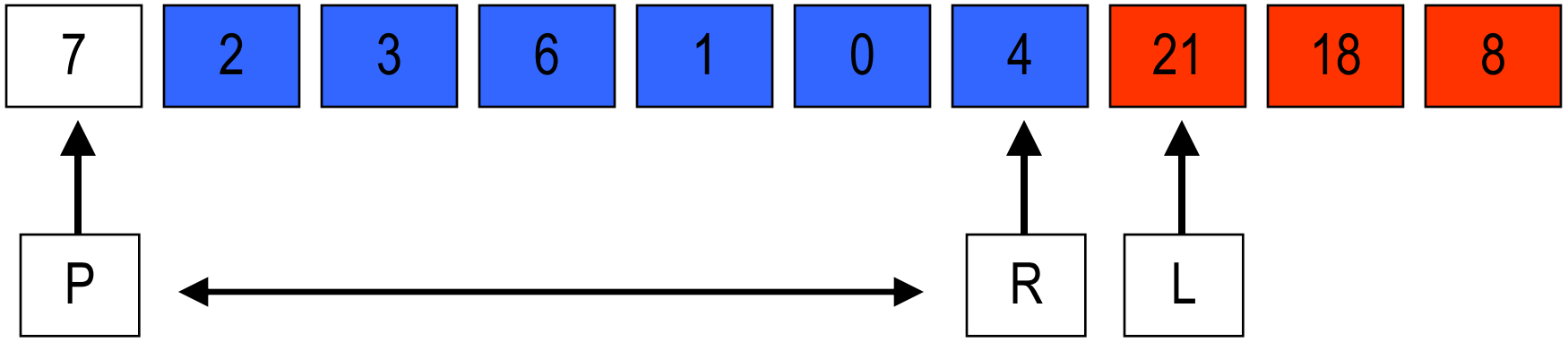


Nun wandert der rechte Pointer zum nächsten „blauen Wert“.

Funktionsweise D13

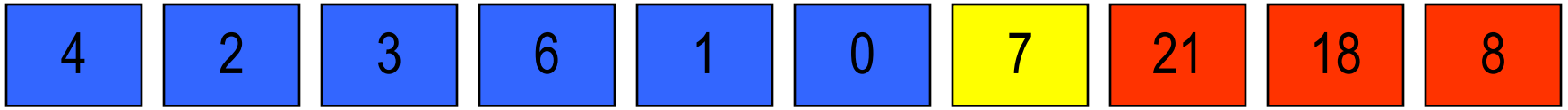


Der rechte Pointer zeigt nun auf einen blauen Wert.

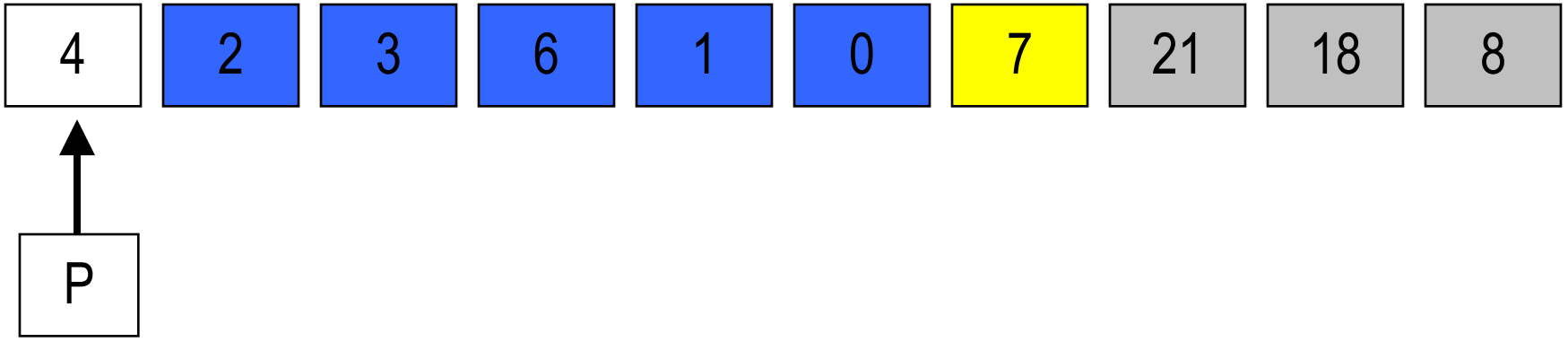


Die beiden Pointer haben sich „gekreuzt“, deshalb wird der Wert des rechten Pointers mit dem Pivotelement vertauscht.

Funktionsweise D15



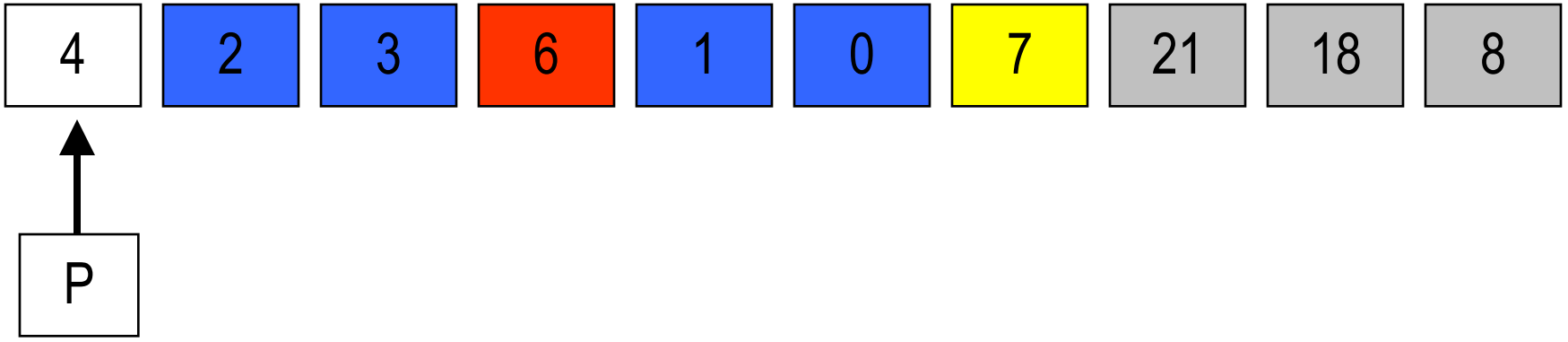
Die Zahl 7 ist bereits an ihrem endgültigen Platz eingeordnet worden. Für die beiden noch unsortierten Bereiche links und rechts wird der Algorithmus mit neuen Parametern wieder aufgerufen.



Wir bearbeiten hier zuerst den linken Teil.

Das Pivotelement wird neu gewählt.

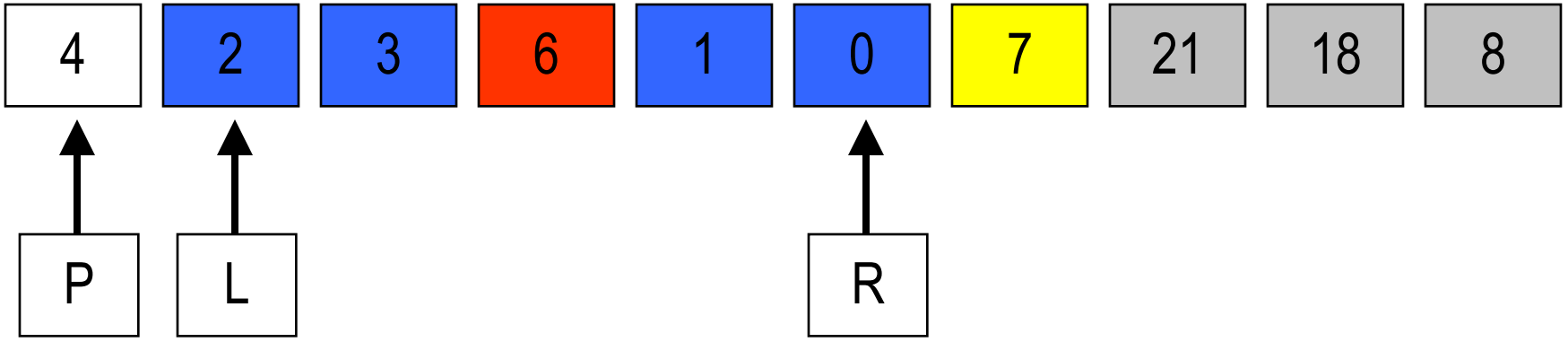
Funktionsweise D17



Den größeren Elementen wird wieder die Farbe rot, den kleineren die Farbe blau zugeordnet.

Die grauen Elemente spielen im Moment keine Rolle.

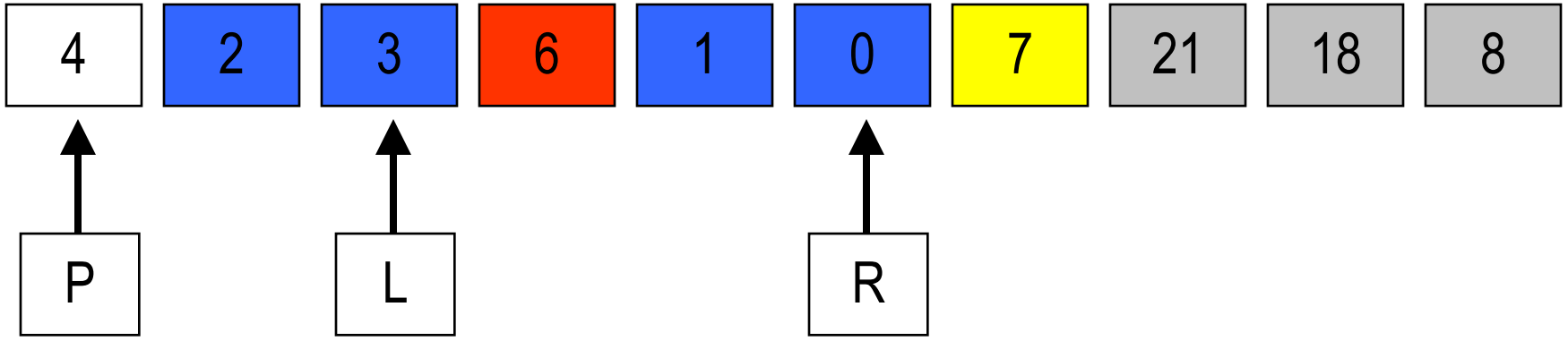
Funktionsweise D18



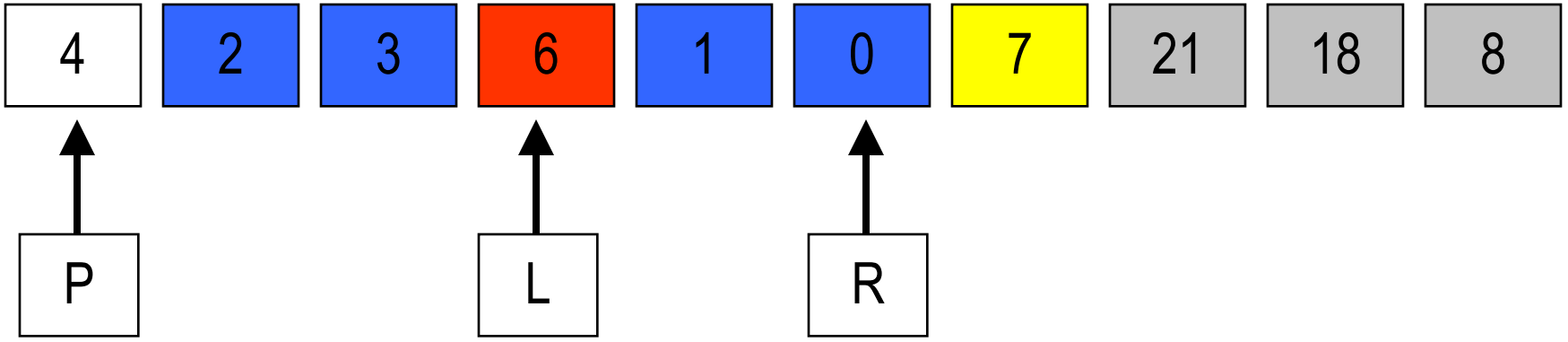
Die Pointer werden auf die Grenzen des zu sortierenden Bereich gesetzt.

Zuerst lassen wir wieder den linken Pointer bis zum ersten „roten Element“ wandern.

Funktionsweise D19

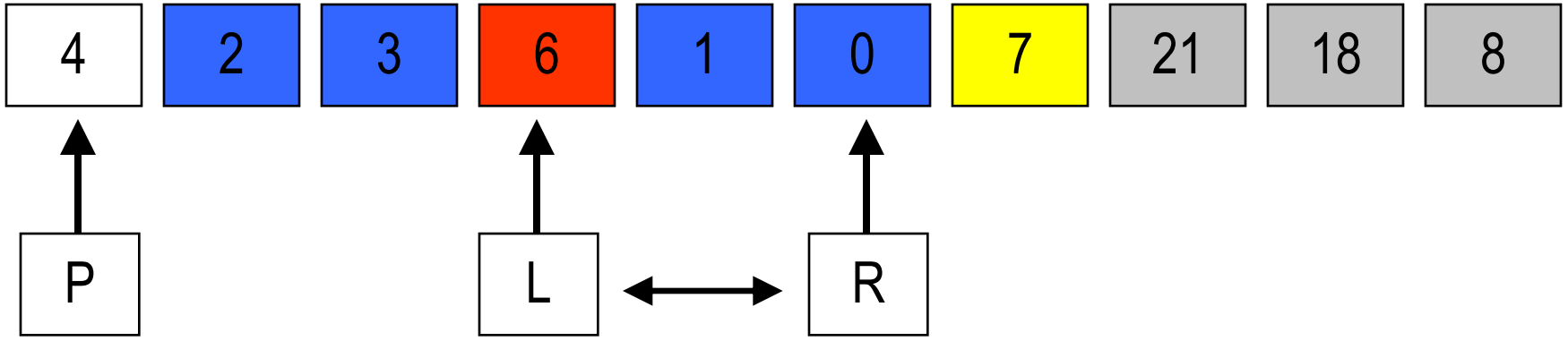


Funktionsweise D20



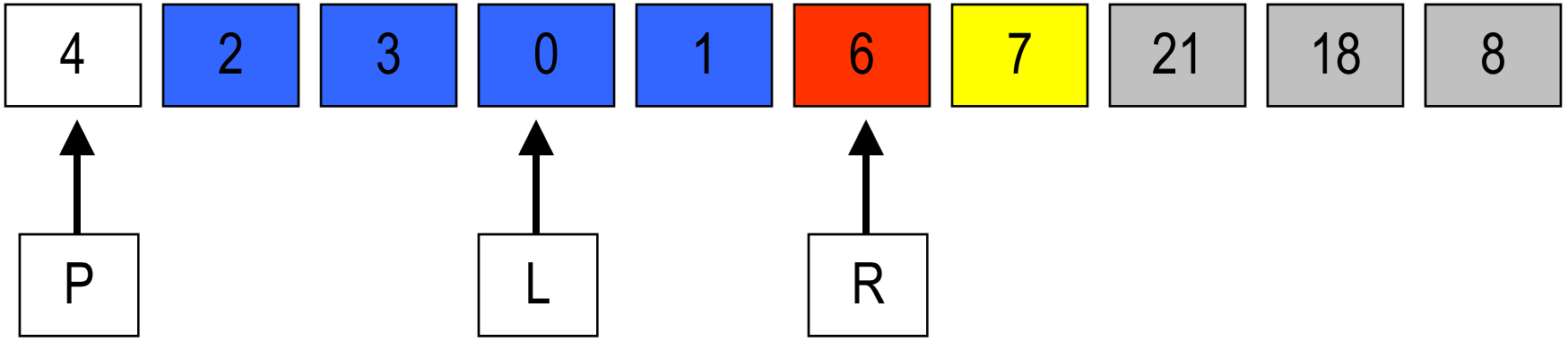
Der rechte Pointer zeigt bereits auf ein „blaues Element“.

Funktionsweise D21



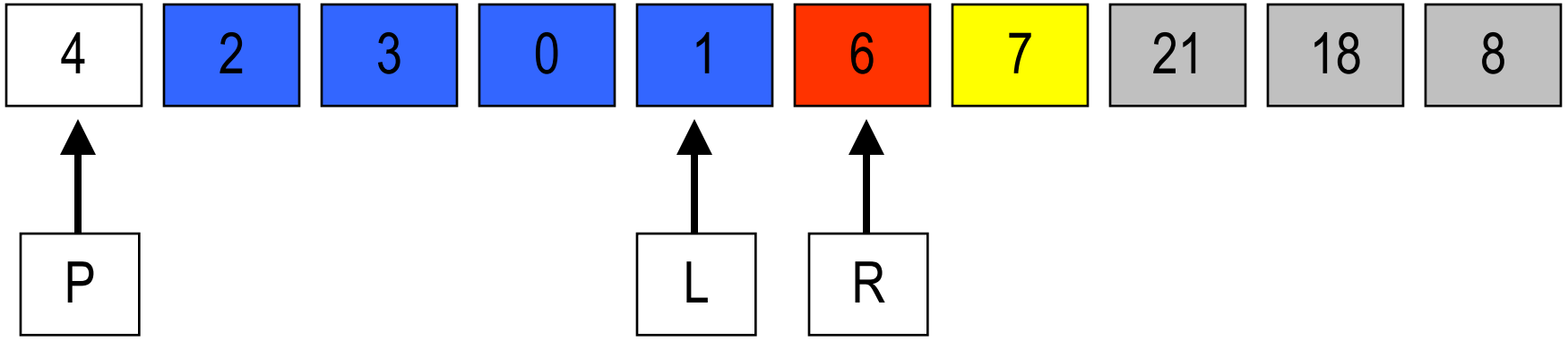
Wir vertauschen die Werte, auf die die Pointer zeigen.

Funktionsweise D22

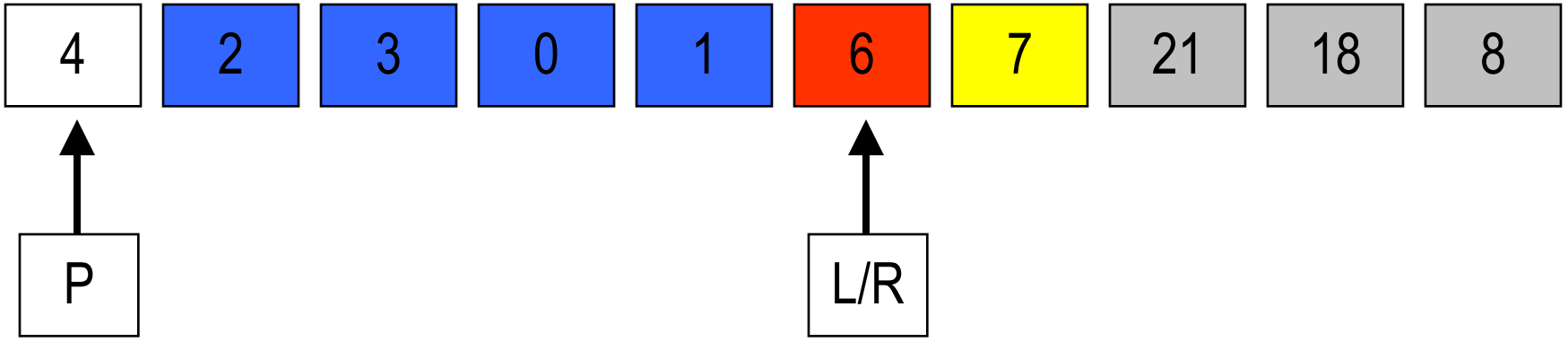


Der linke Pointer wandert weiter zum nächsten „roten Element“.

Funktionsweise D23

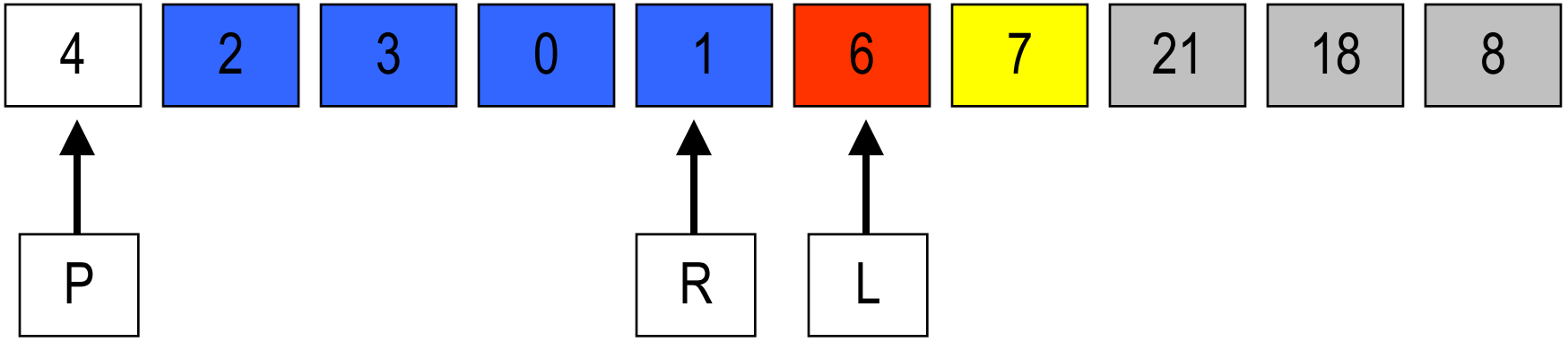


Funktionsweise D24



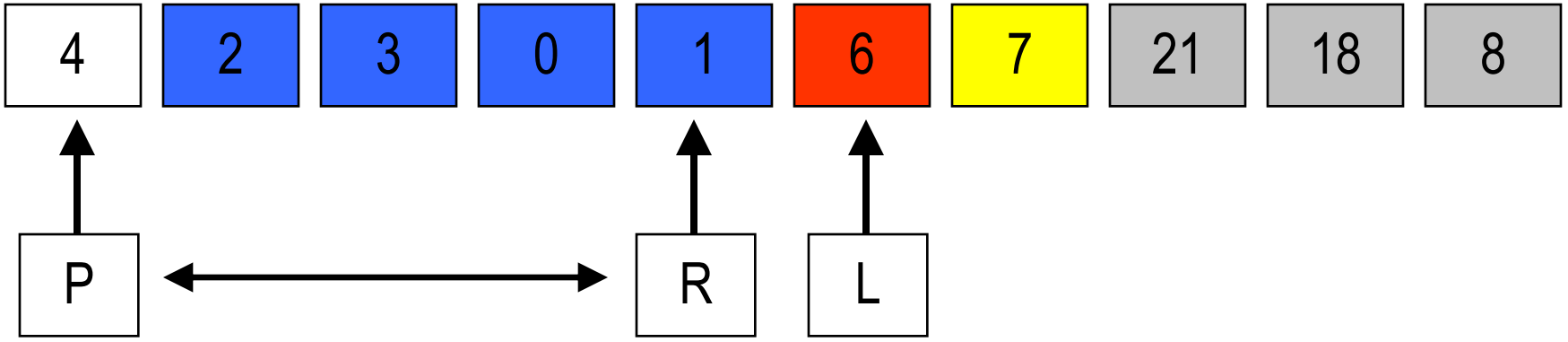
Der linke Pointer zeigt nun auf ein „rotes Element“. Wir lassen den rechten Pointer zum nächsten „blauen Element“ wandern.

Funktionsweise D25



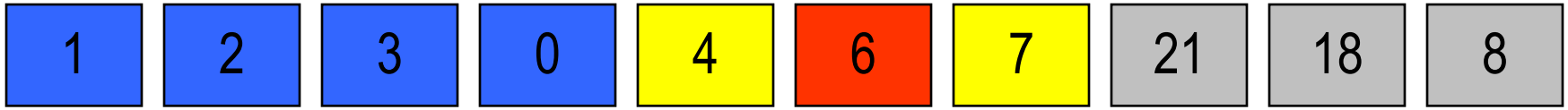
Der rechte Pointer zeigt auf ein „blaues Element“.

Funktionsweise D26



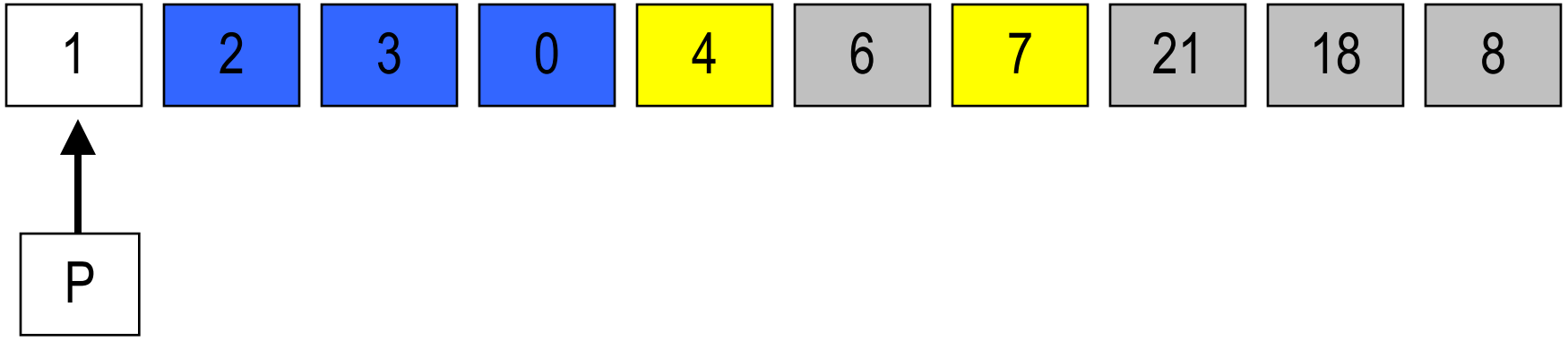
Da der Index des rechten Pointer kleiner als der des linken ist, vertauschen wir das Pivotelement mit dem Wert auf den der rechte Pointer zeigt.

Funktionsweise D27



Die Zahl 4 befindet sich nun bereits auf ihrem endgültigen Platz.
Der Algorithmus wird erneut aufgerufen.

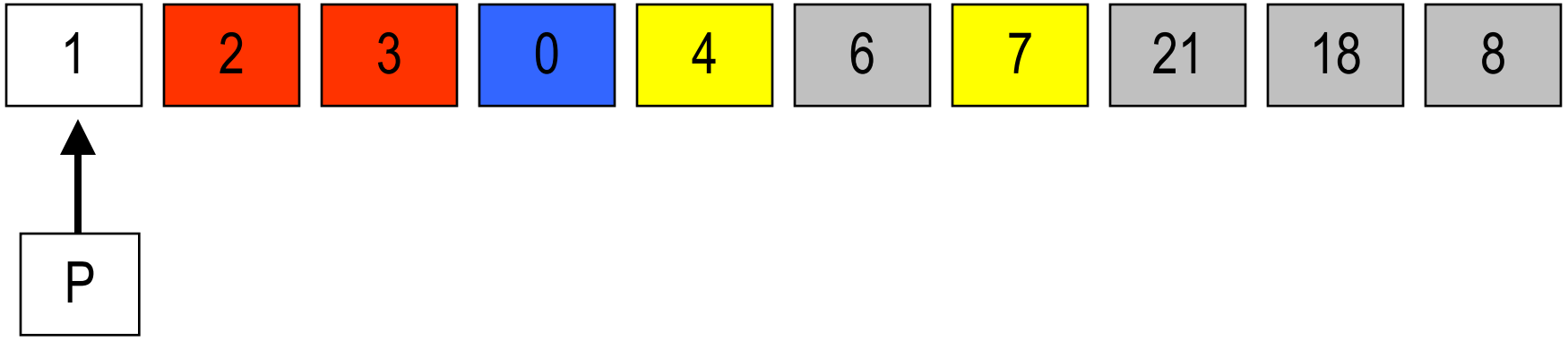
Funktionsweise D28



Wir sortieren wiederum zuerst den Teil links vom Wert 4.

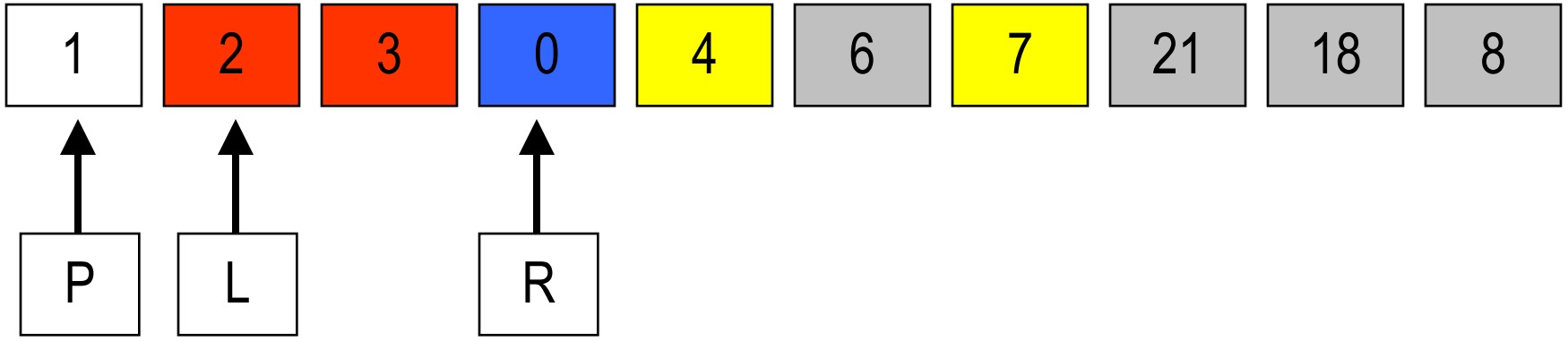
Zuerst wählen wir das Pivotelement.

Funktionsweise D29



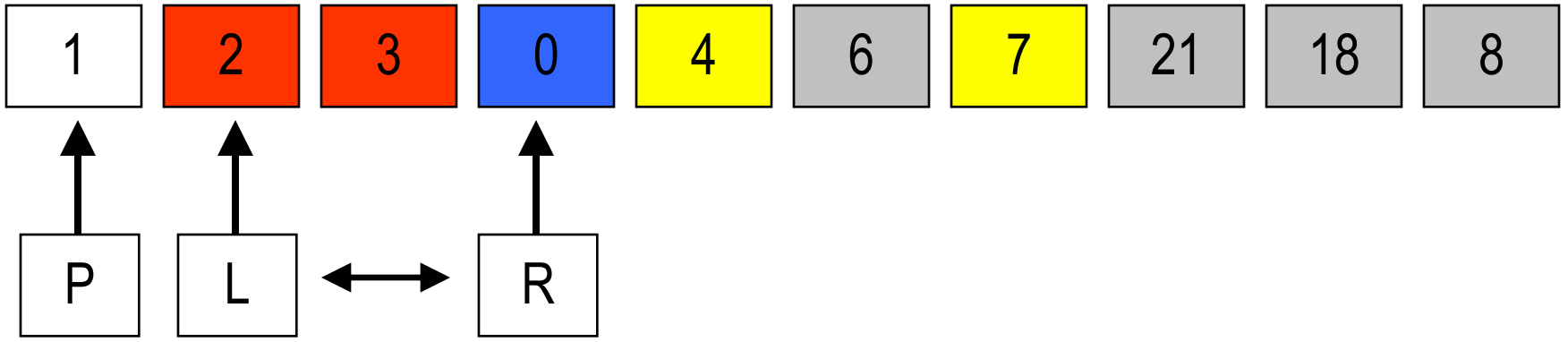
Die Elemente werden eingefärbt.

Funktionsweise D30



Die Pointer werden an die Grenzen des Bereichs gesetzt.

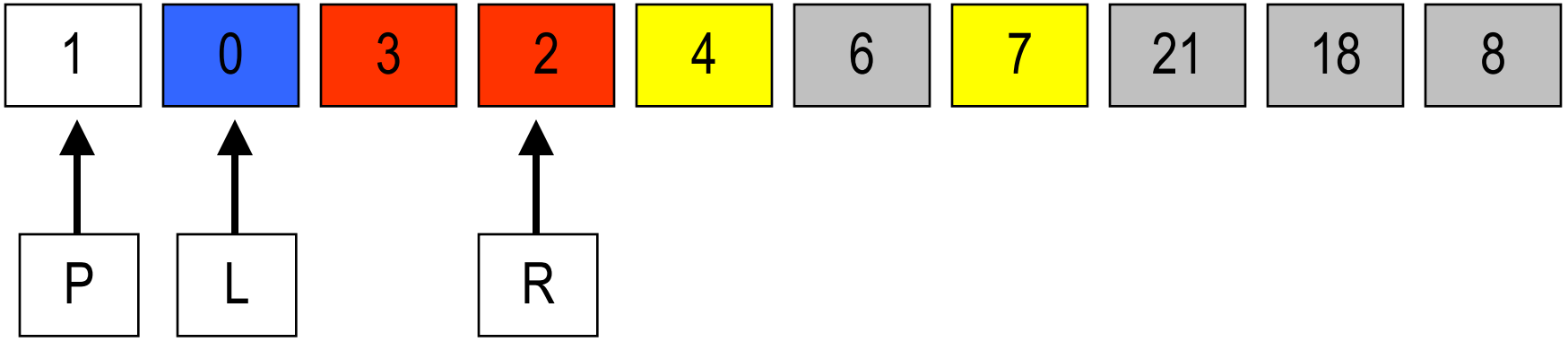
Funktionsweise D31



Der linke Pointer zeigt bereits auf ein „rotes“, der Rechte bereits auf ein „blaues“ Element.

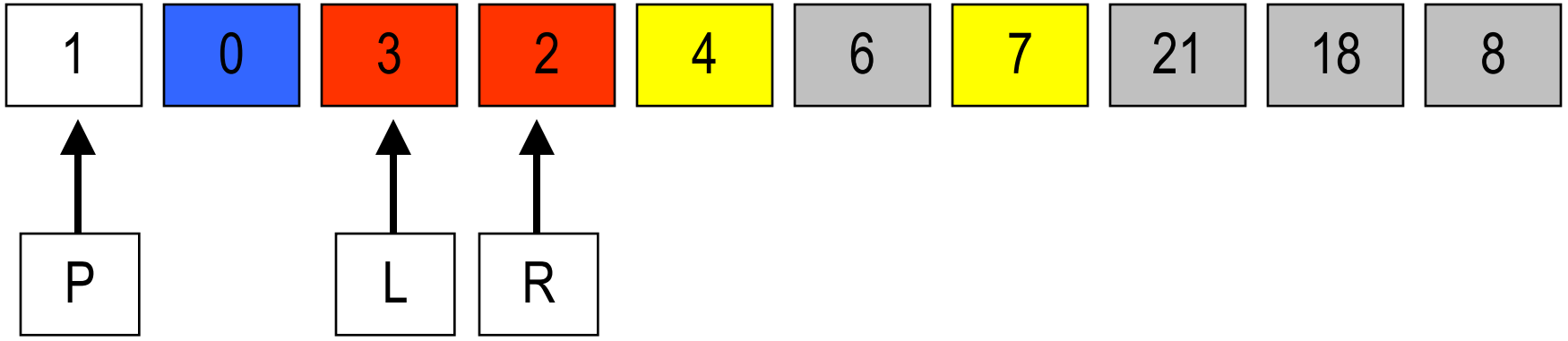
Die beiden Elemente werden vertauscht.

Funktionsweise D32



Nun lassen wir wieder den linken Pointer bis zum nächsten „roten“ Wert wandern.

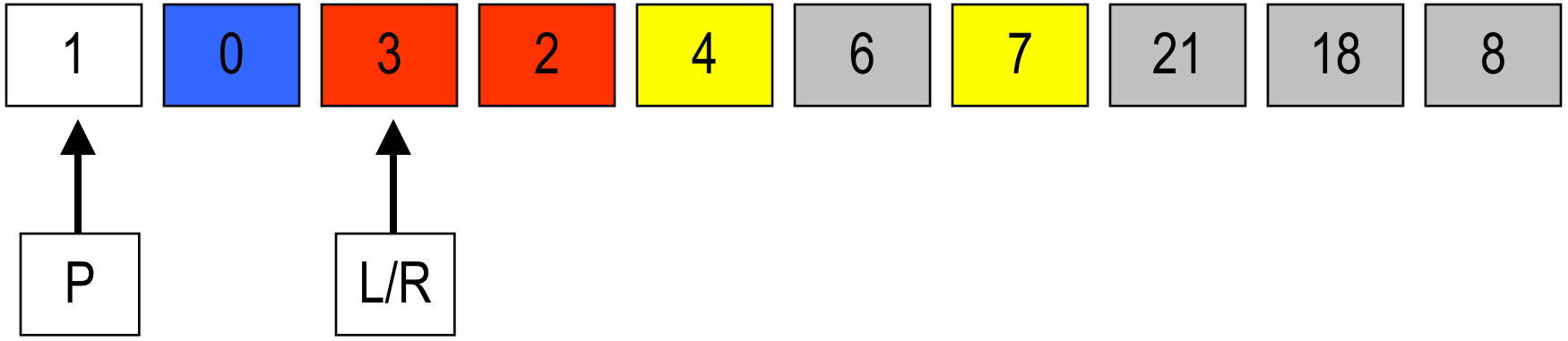
Funktionsweise D33



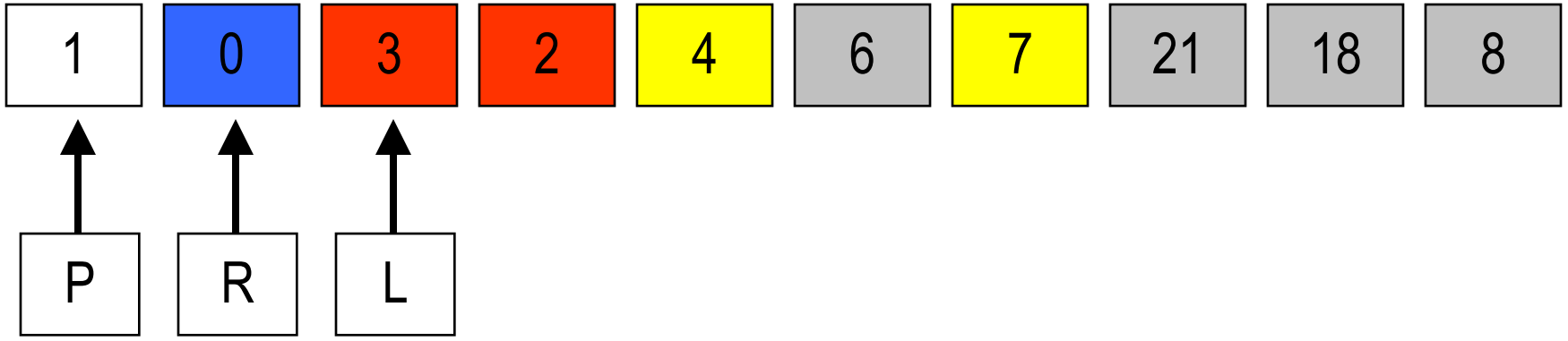
Der linke Pointer zeigt auf einen Wert, der größer als das Pivotelement ist.

Nun lassen wir den rechten Pointer zum nächsten „blauen Element“ wandern.

Funktionsweise D34

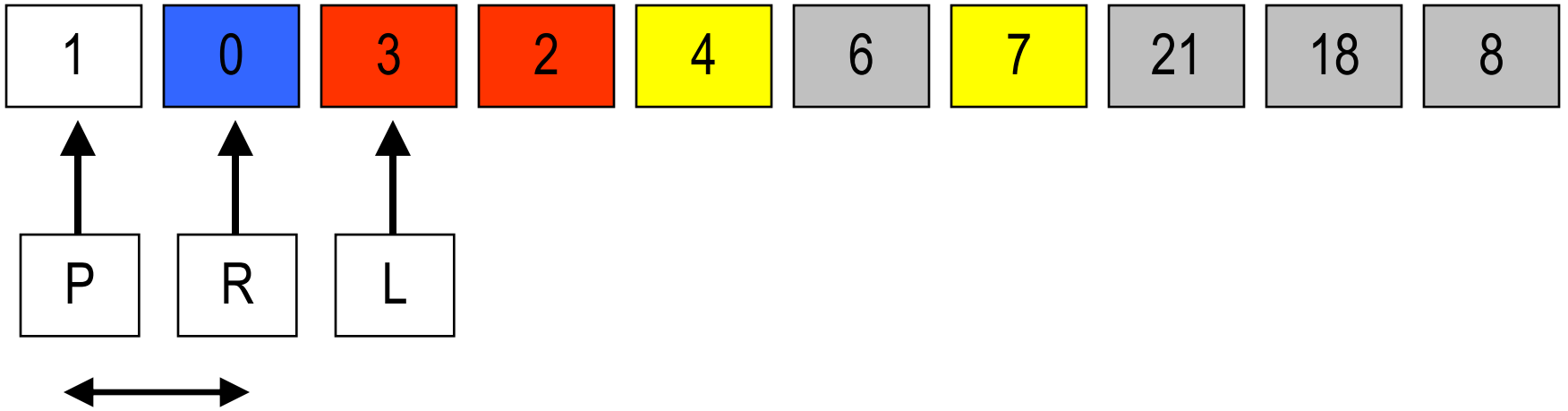


Funktionsweise D35



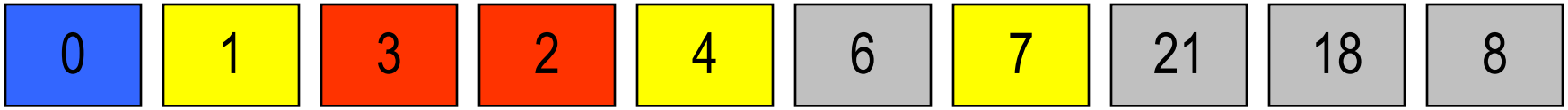
Der rechte Pointer zeigt auf ein blaues Element.

Funktionsweise D36



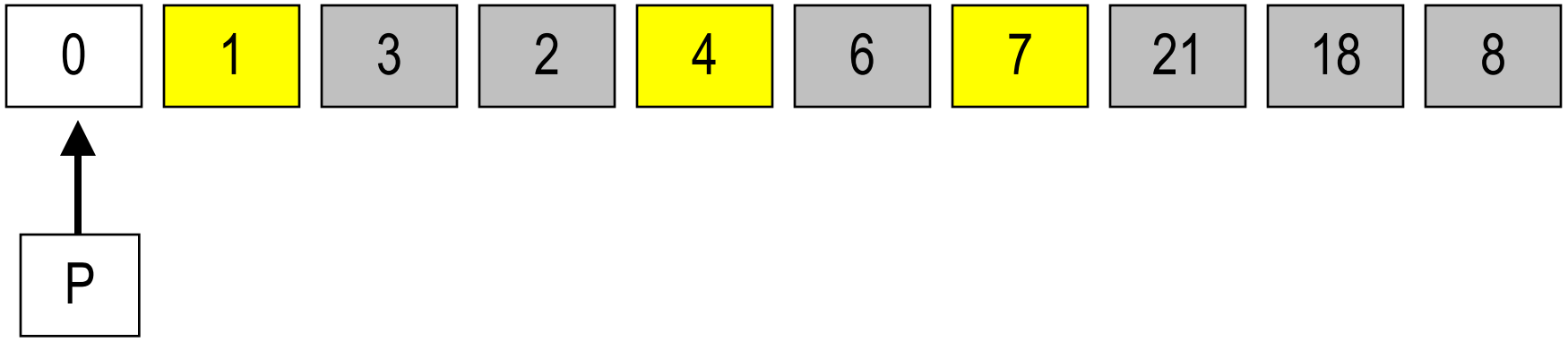
Da sich die beiden Pointer „gekreuzt“ haben, wird der Wert auf den der rechte Pointer zeigt mit dem Pivotelement vertauscht.

Funktionsweise D37



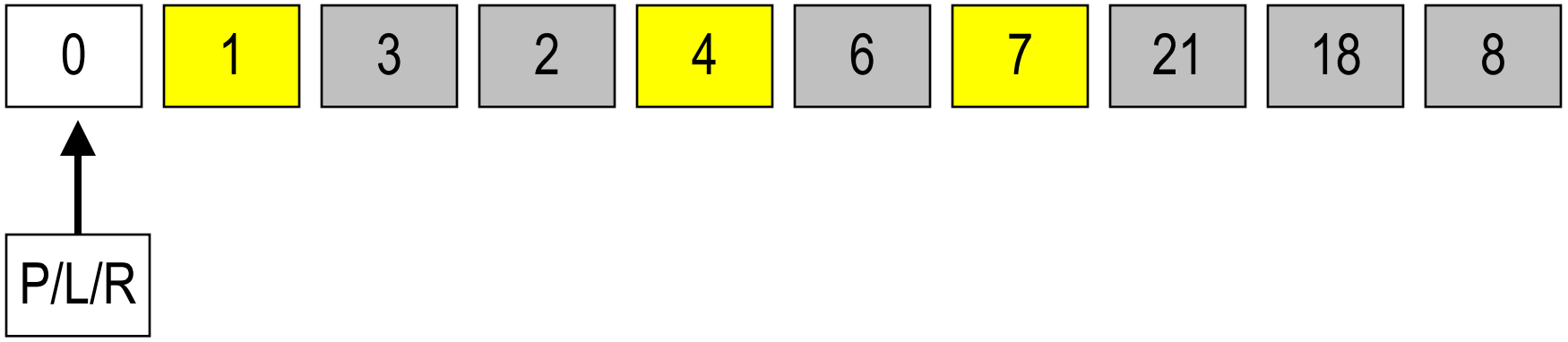
Der Wert 1 befindet sich bereits an der richtigen Stelle.

Funktionsweise D38



Das Pivotelement wird neu bestimmt.

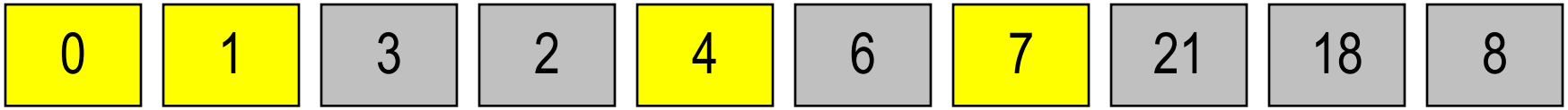
Funktionsweise D39



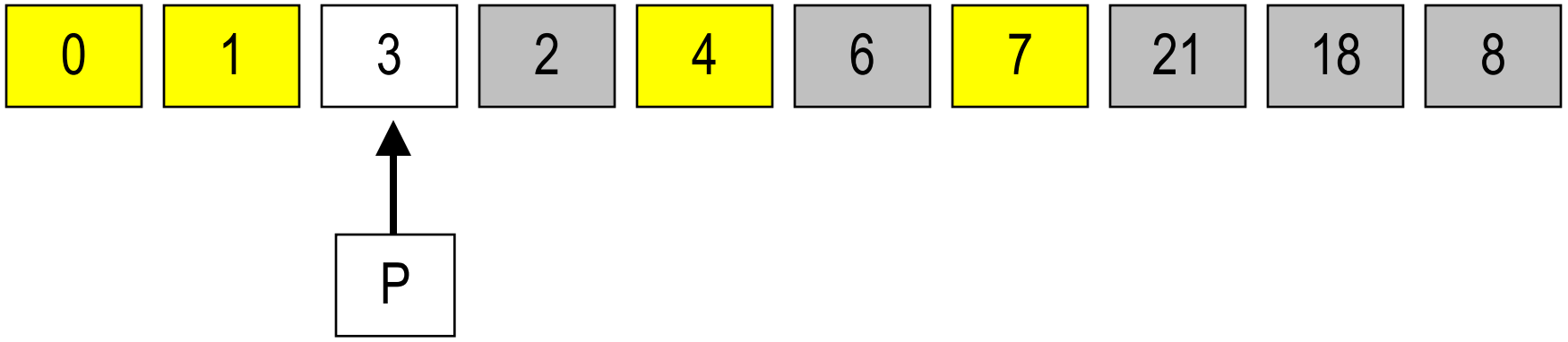
Die Pointer werden an die Grenzen des zu sortierenden Bereichs gesetzt.

Da die Pointer zu Beginn auf das gleiche Element zeigen, wird der Algorithmus nicht mehr aufgerufen. Der Wert ist bereits sortiert.

Funktionsweise D40



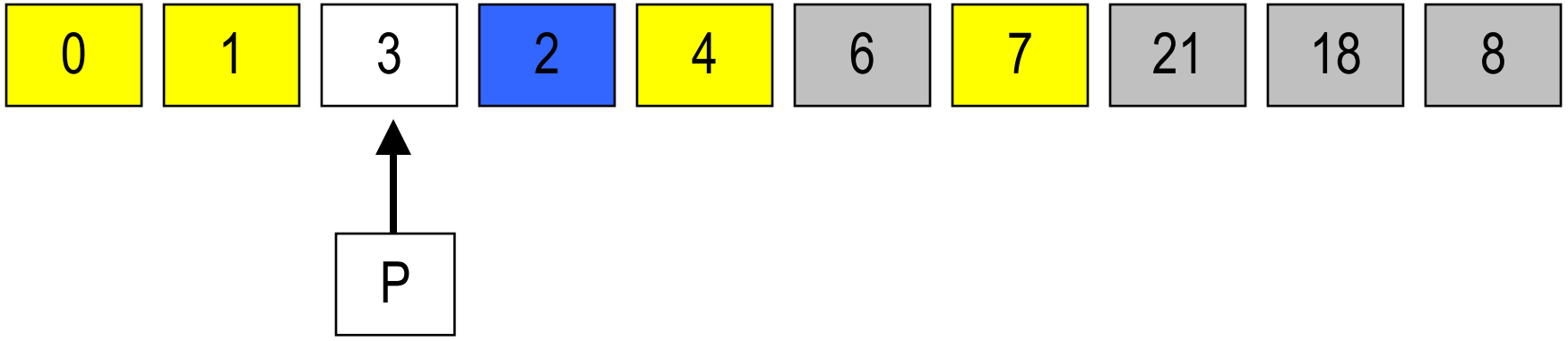
Der Wert 0 befindet sich an der richtigen Stelle.



Das Array wird nun weiter abgearbeitet.

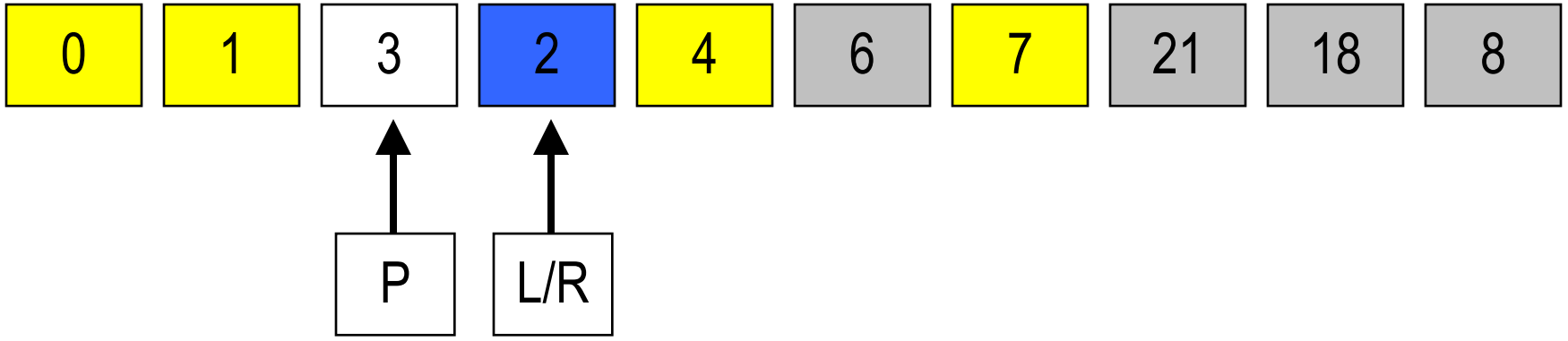
Zuerst bestimmen wir das neue Pivotelement.

Funktionsweise D42



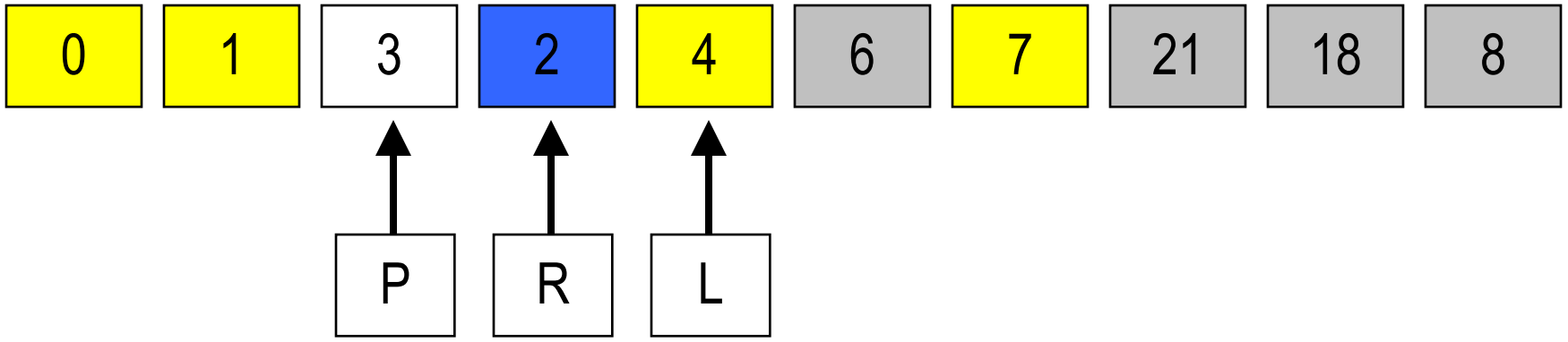
Die Elemente werden ihrem Wert entsprechend eingefärbt.

Funktionsweise D43



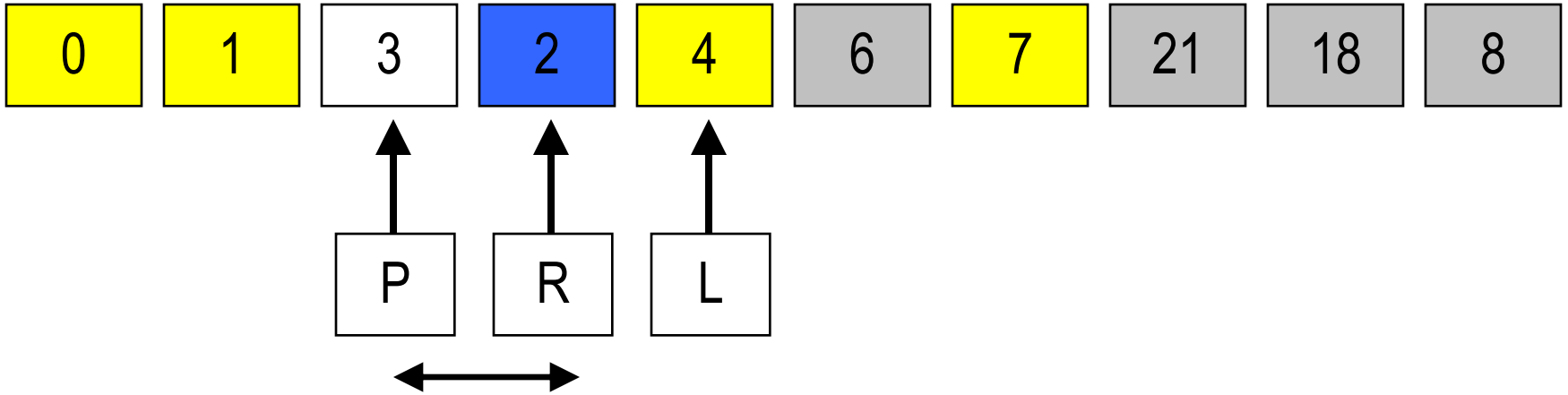
Die Pointer werden an die Ränder gesetzt.

Der linke Pointer begibt sich auf die Suche nach dem nächsten Element das größer als das Pivotelement ist (oder gleich groß ist).



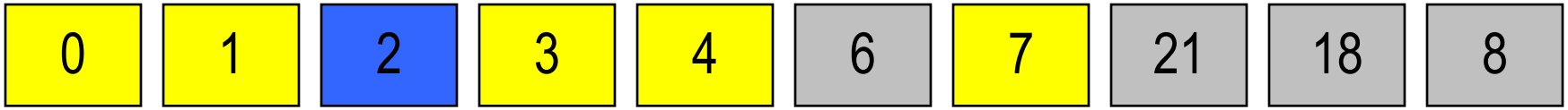
Der linke Pointer zeigt nun auf den Wert 4. Dieser befindet sich außerhalb des zu sortierenden Bereichs. Dies bedeutet, dass sich im zu sortierenden Bereich kein Element befindet das größer als das Pivotelement ist.

Funktionsweise D45



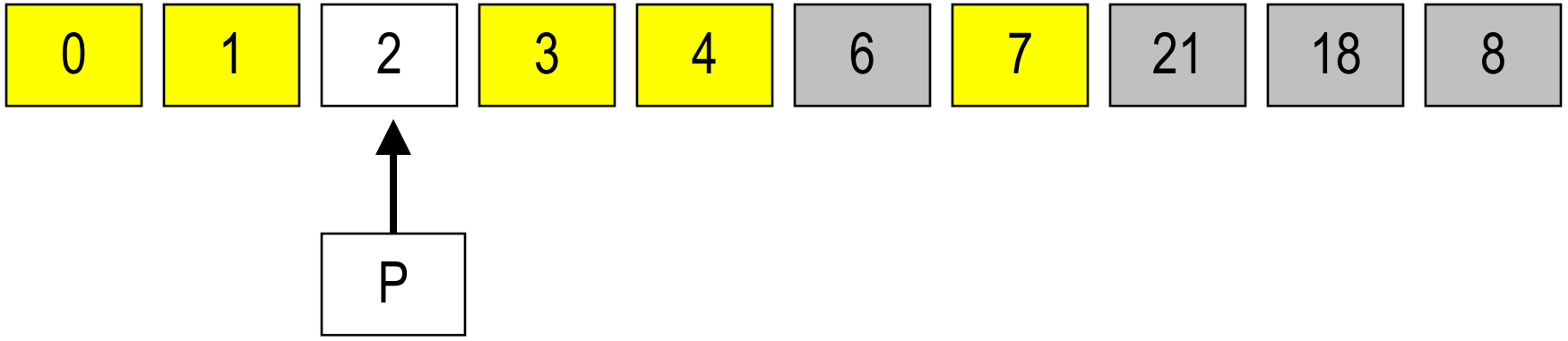
Der rechte Pointer zeigt bereits auf ein blaues Element. Da sich die Pointer „gekreuzt“ haben, wird das Pivotelement mit dem Element auf das R zeigt vertauscht.

Funktionsweise D46



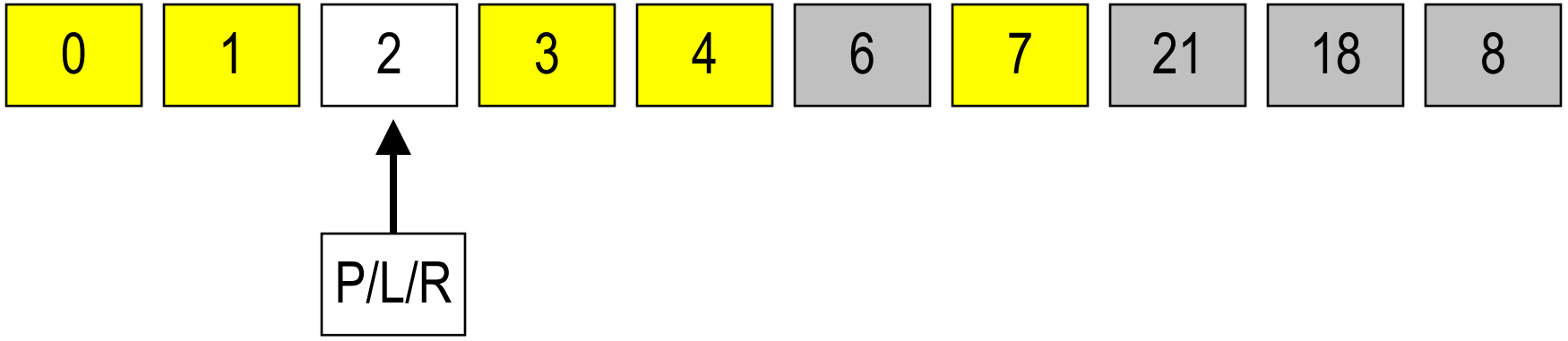
3 befindet sich nun an der richtigen Stelle.

Funktionsweise D47



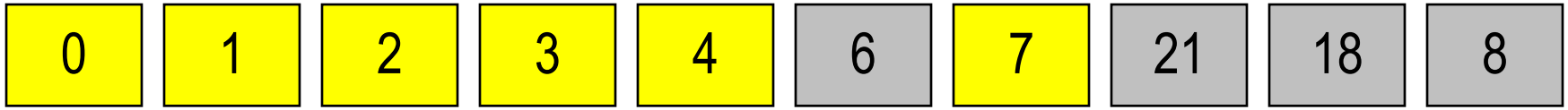
Das Pivotelement wird neu bestimmt.

Funktionsweise D48



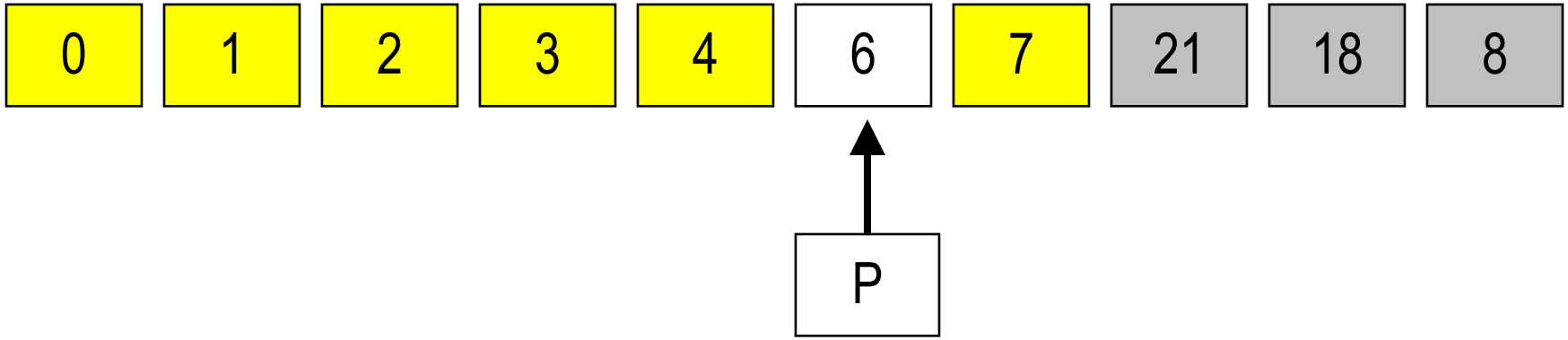
Die Pointer werden an die Grenzen des zu sortierenden Bereichs gesetzt.

Da die Pointer zu Beginn auf das gleiche Element zeigen, wird der Algorithmus nicht mehr aufgerufen. Der Wert ist bereits sortiert.



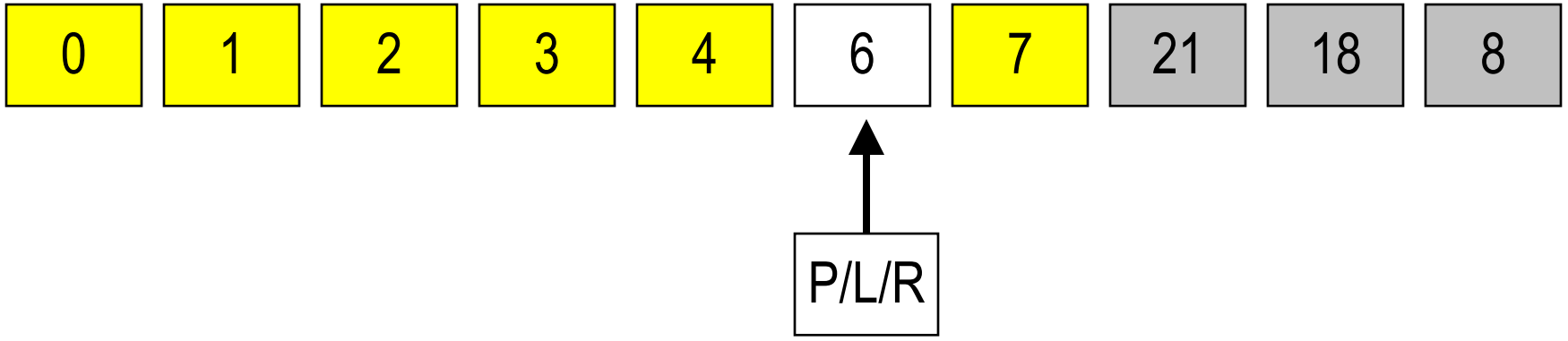
2 befindet sich nun an der richtigen Stelle.

Funktionsweise D50



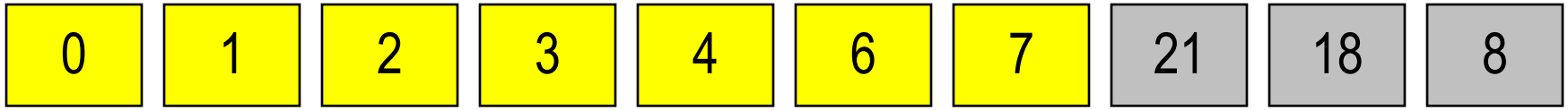
Das Pivotelement wird neu bestimmt.

Funktionsweise D51



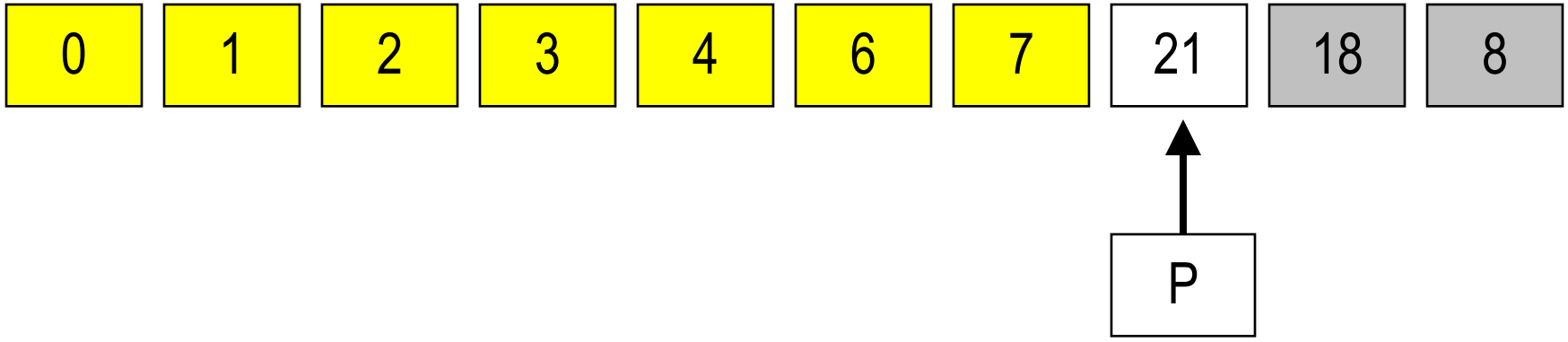
Die Pointer werden an die Grenzen des zu sortierenden Bereichs gesetzt.

Da die Pointer zu Beginn auf das gleiche Element zeigen, wird der Algorithmus nicht mehr aufgerufen. Der Wert ist bereits sortiert.



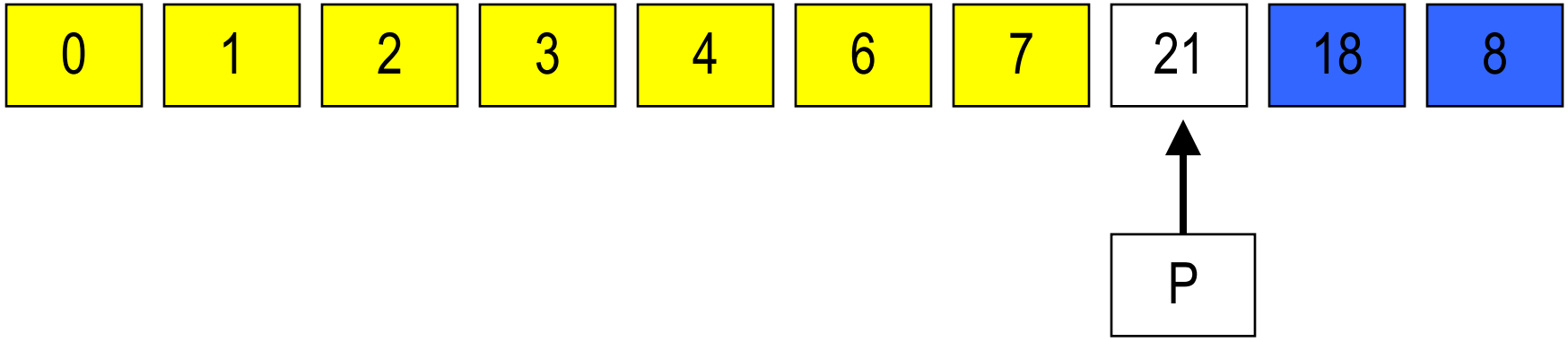
6 ist nun richtig eingeordnet.

Funktionsweise D53



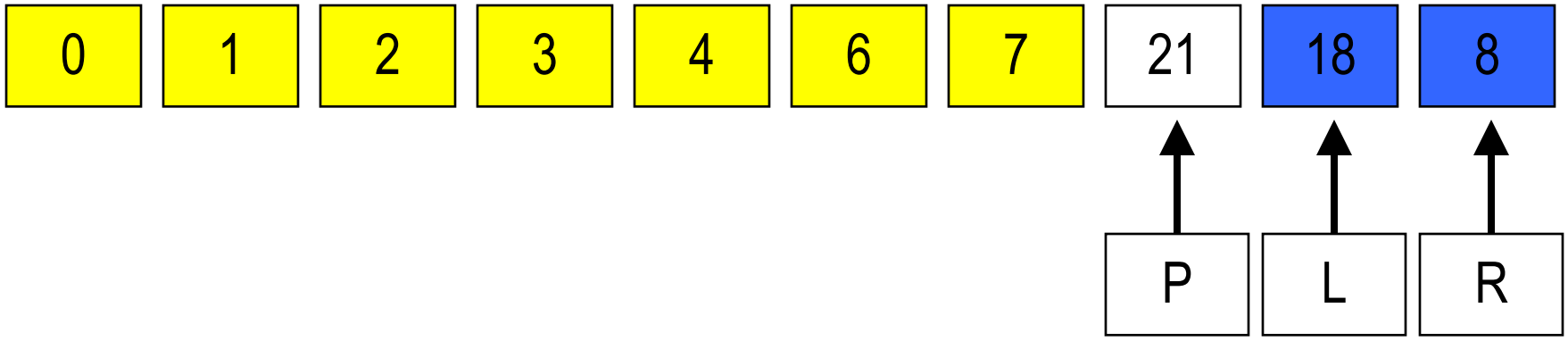
Das Pivotelement wird neu gewählt.

Funktionsweise D54



Die Elemente werden entsprechend ihrem Wert (blau = kleiner, rot = größer) in Bezug auf das Pivotelement eingefärbt.

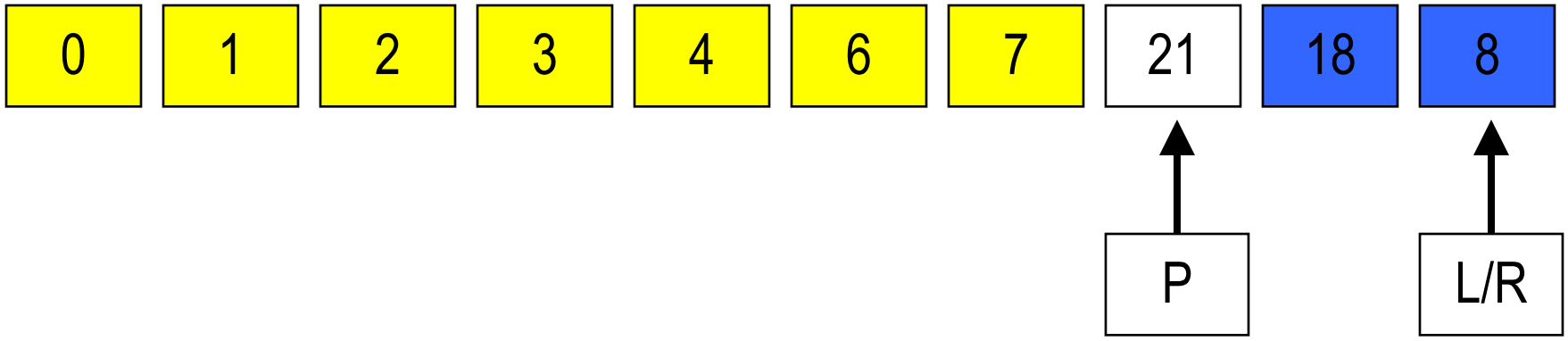
Funktionsweise D55



Die Pointer werden an die Ränder gesetzt.

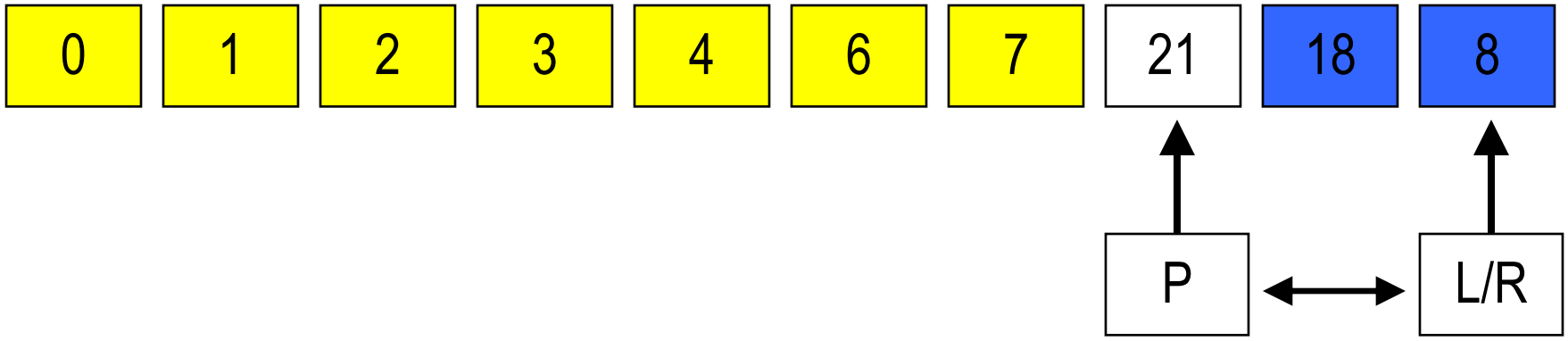
Der linke Pointer wandert nach rechts bis zum nächsten „roten“ Element.

Funktionsweise D56

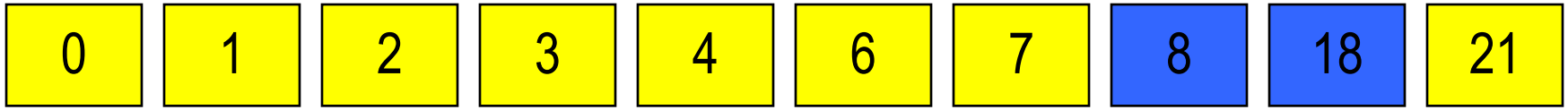


Da sich kein „rotes“ Element im Bereich findet, zeigt er schlussendlich auf das am weitesten rechts stehende blaue Element.

Funktionsweise D57

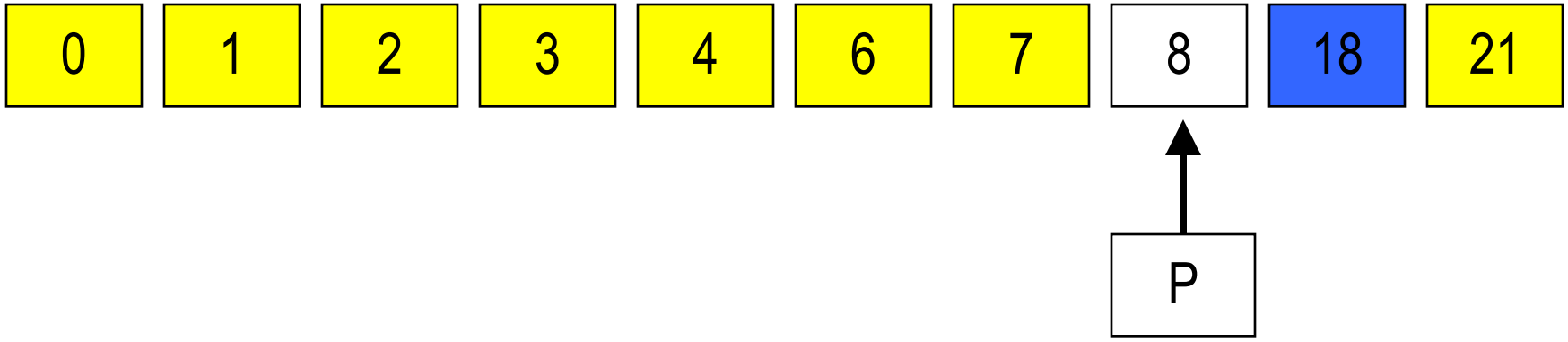


Der rechte Pointer zeigt bereits auf ein blaues Element. Da sich die beiden Pointer „gekreuzt“ haben, wird der Wert des rechten Pointers mit dem Pivotelement vertauscht.



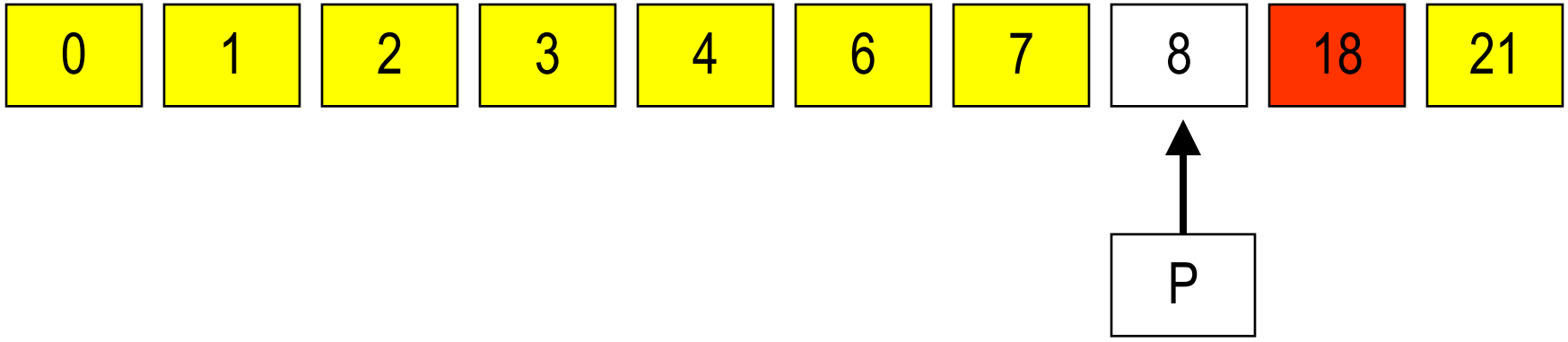
21 ist nun korrekt eingeordnet.

Funktionsweise D59



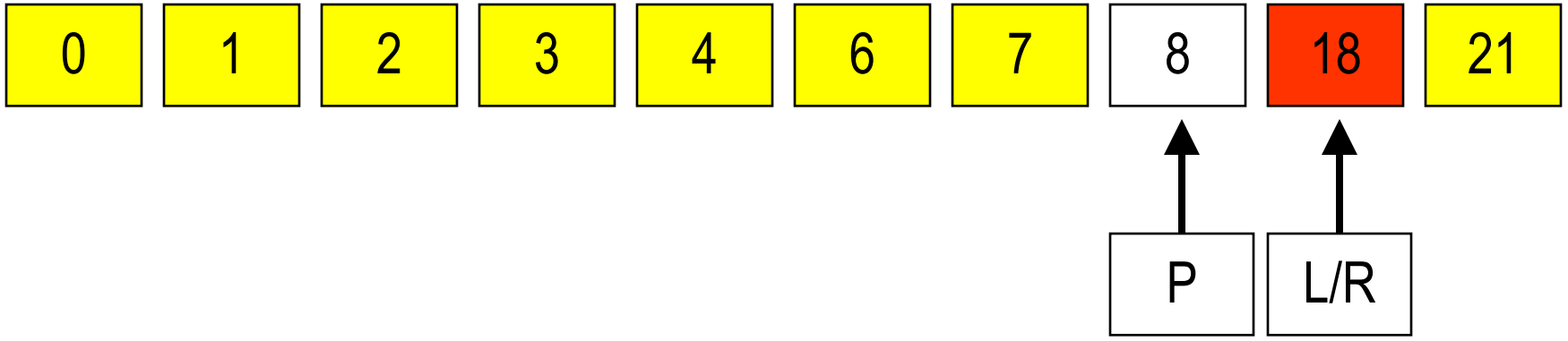
Das Pivotelement wird neu gewählt.

Funktionsweise D60



Die Elemente werden entsprechend eingefärbt.

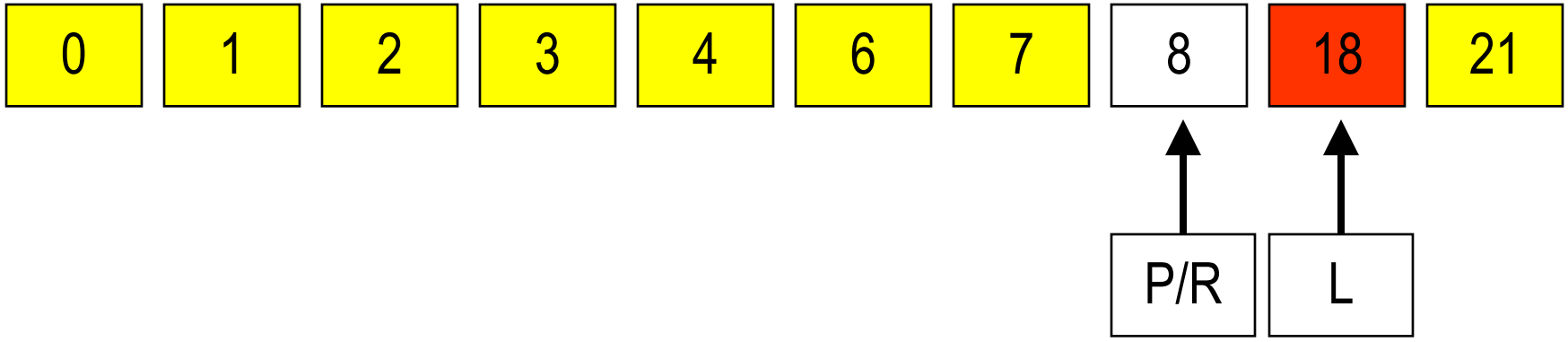
Funktionsweise D61



Die Pointer werden gesetzt.

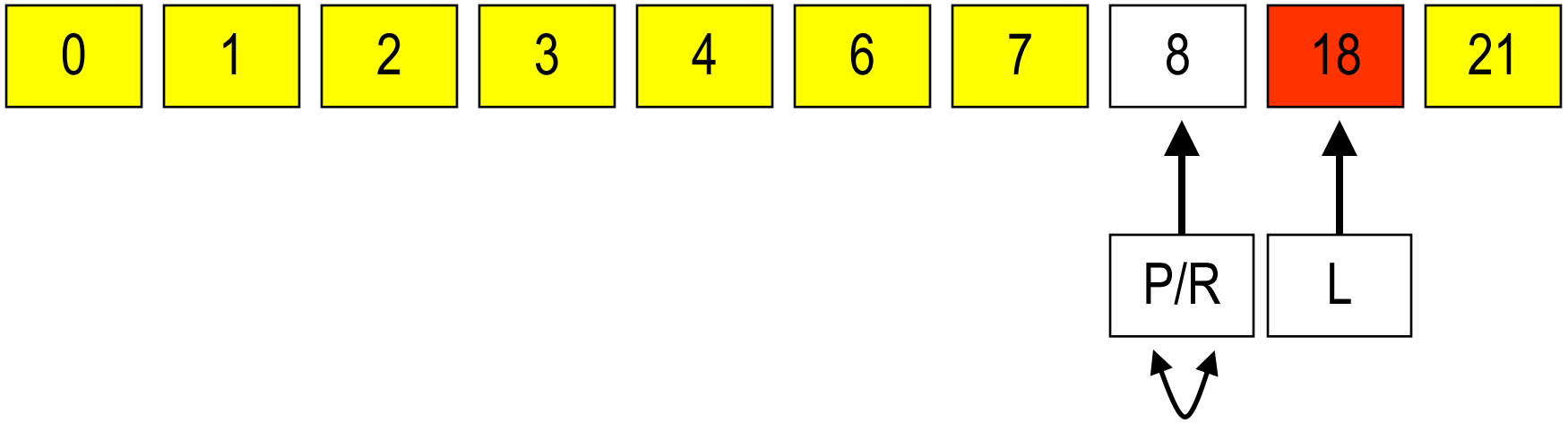
Der linke Pointer zeigt bereits auf ein „rotes Element“.

Der rechte wandert bis zum nächsten Element, das kleiner als oder gleich groß wie das Pivotelement ist.



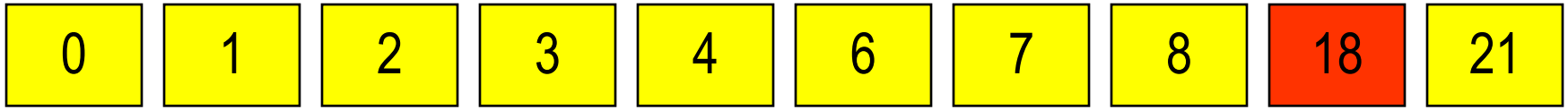
In diesem Fall zeigt der rechte Pointer schlussendlich auf das Pivotelement selbst.

Funktionsweise D63



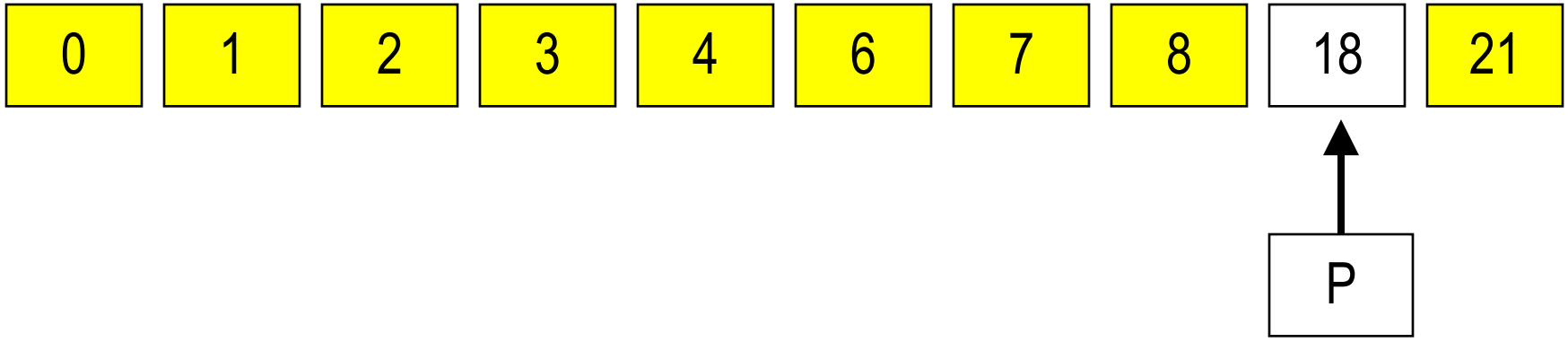
Das Pivotelement wird „mit sich selbst“ vertauscht.

Funktionsweise D64



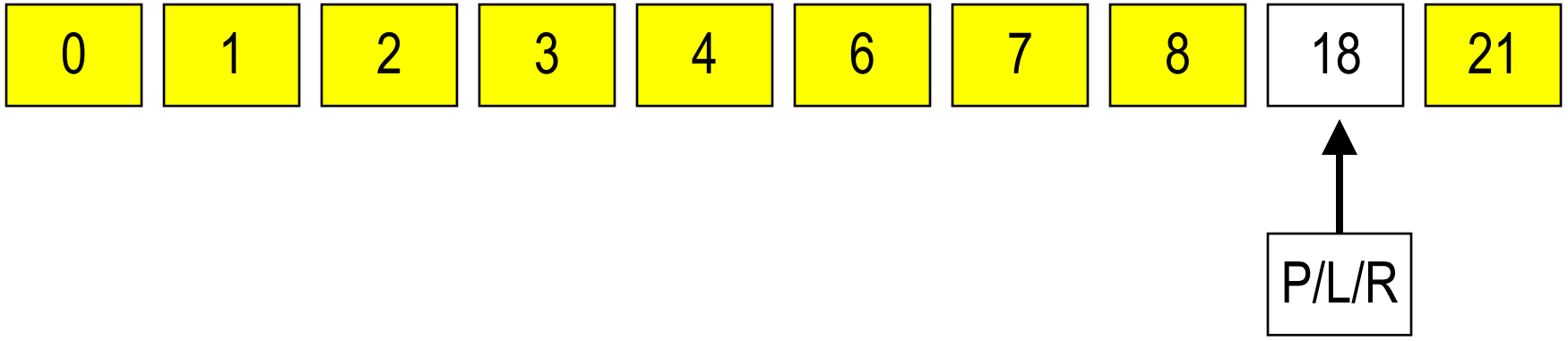
8 ist nun korrekt eingeordnet.

Funktionsweise D65



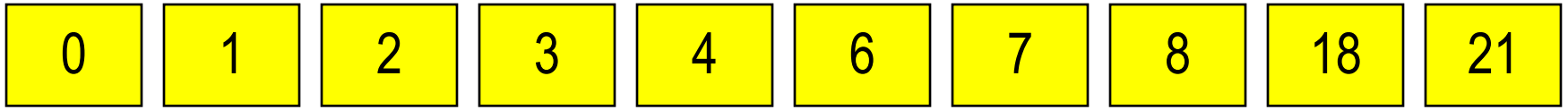
Das Pivotelement wird neu gewählt.

Funktionsweise D66



Die Pointer werden an die Grenzen des zu sortierenden Bereichs gesetzt.

Da die Pointer zu Beginn auf das gleiche Element zeigen, wird der Algorithmus nicht mehr aufgerufen. Der Wert ist bereits sortiert.



Das Array ist nun korrekt sortiert.