

# Aufzugalgorithmus



# Lesen und Schreiben von HDDs (Fortsetzung)

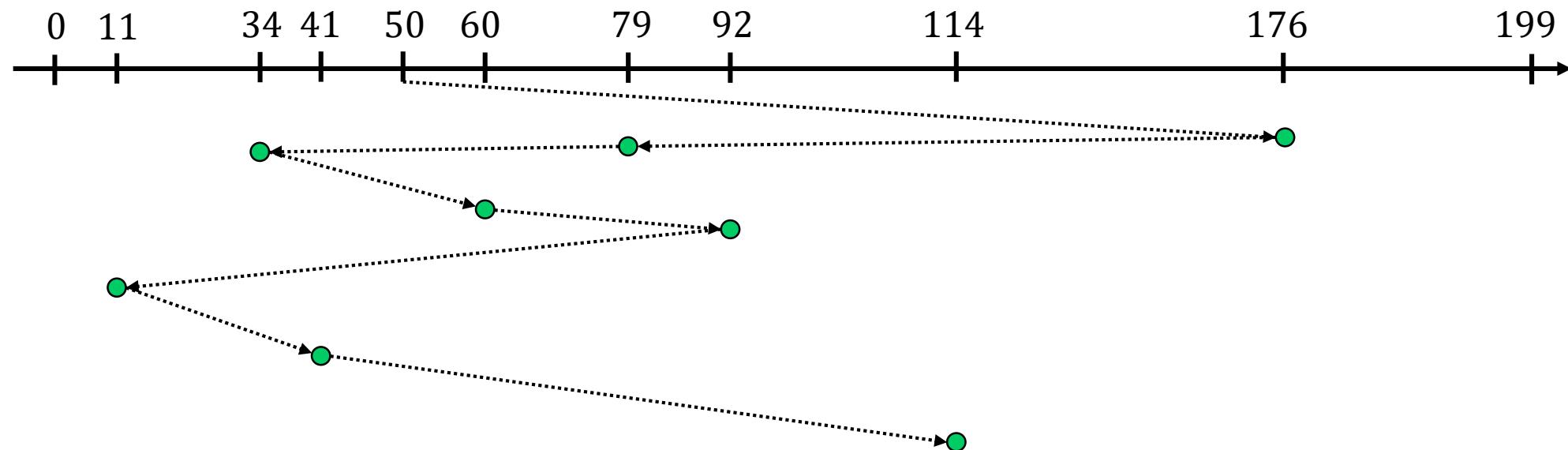
- Lese- und Schreibvorgänge beanspruchen bei HDDs wesentlich mehr Zeit als bei Festkörperspeichern (SSD)
- Ständige **Spurwechsel** sollten daher unbedingt vermieden werden, um wertvolle Zeit zu sparen
- Analogie: Automatischer Aufzug über mehrere Etagen

# Aufzugalgorithmus für Anfragemengen

- 1) **Ausgangssituation:** Der Aufzug befindet sich in einer bestimmten Etage. Es gibt eine Menge  $L$  von Anfragen für verschiedene Etagen. Der Aufzug hat eine aktuelle Bewegungsrichtung (aufwärts  $\uparrow = 1$  oder abwärts  $\downarrow = -1$ ).
- 2) **Bewegungsrichtung festlegen:** Wenn der Aufzug still steht, wähle die Richtung zur nächsten Anfrage. Ansonsten behalte die aktuelle Richtung bei.
- 3) **Anfragen sortieren:** Sortiere die Anfragen in der aktuellen Richtung.
- 4) **Abarbeiten der Anfragen:** Bewege den Aufzug in der festgelegten Richtung. Halte an jeder Etage, für die eine Anfrage vorliegt. Nimm neue Anfragen in derselben Richtung auf.
- 5) **Richtungswechsel:** Wenn keine weiteren Anfragen in der aktuellen Richtung vorliegen, wechsle die Richtung. Beginne wieder mit Schritt 3.
- 6) **Leerlauf:** Wenn keine Anfragen mehr vorliegen, warte auf neue Anfragen. Bei einer neuen Anfrage, beginne wieder mit Schritt 2.

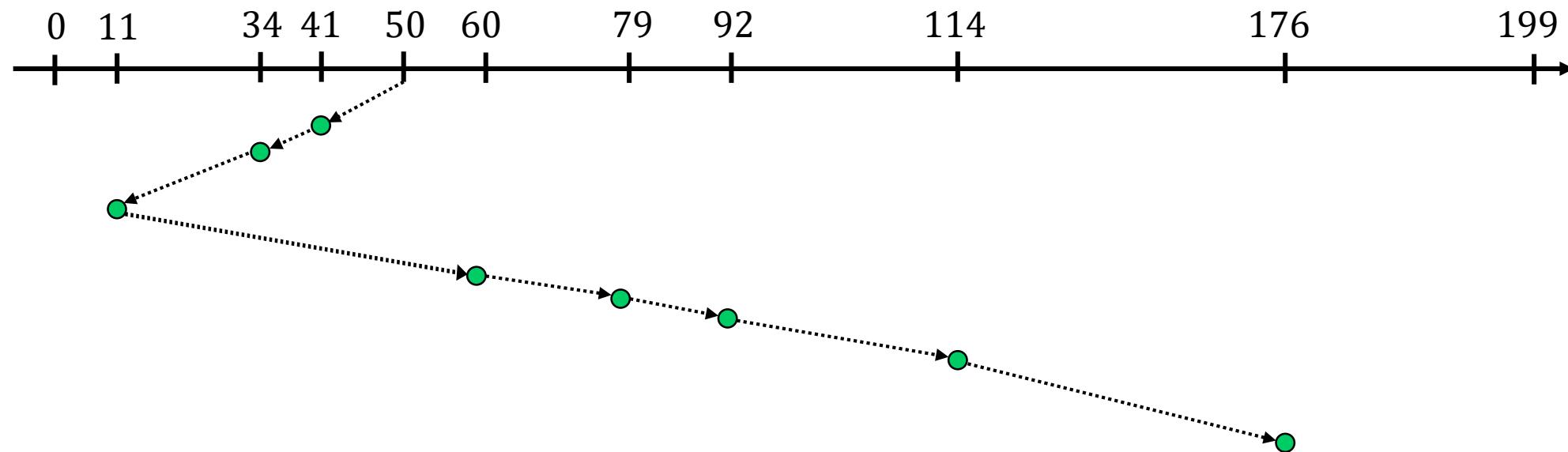
# HDD-Scheduling nach Windhundprinzip (FCFS)

$$L_0 = \{176, 79, 34, 41, 60, 92, 11, 41, 114\}$$



# HDD-Scheduling nach Aufzugsalgorithmus (SCAN)

$$L_0 = \{176, 79, 34, 41, 60, 92, 11, 41, 114\}$$



# Aufzugalgorithmus in Echtzeit

- Im realen System ist es möglich, dass im laufenden E/A-Vorgang die Warteschlange um weitere Anfragen ergänzt wird
- Mengen von Anfragen zum Zeitpunkt  $t$  werden mit  $L_t$  notiert

# Aufzugsalgorithmus in Echtzeit (Übungsbeispiel)

$N = 16, pos = 0, dir = \uparrow$

$L_0 = \{3,7,9,15\}, L_1 = \{2,13\}, L_5 = \{0,7,8,12\}$

