SUM

$A = \{a_1, \dots, a_n\} \subseteq \mathbb{N} \quad , \quad M \in \mathbb{N}$

$\exists B \subseteq A$ mit $\sum\limits_{a_i \in B} a_i = M$?

PARTITION

$A = \{a_1, \dots, a_n\} \subseteq \mathbb{N}$

$\exists D \subseteq A$ (Partitionsmenge) mit

$$\sum_{a_i \in D} a_i = \sum_{a_j \in A \setminus D} a_j = \frac{1}{2} \sum_{i=1}^{n} a_i \ ?$$

Es gilt: SUM $\leq$ PARTITION

$A = \{a_1, \dots, a_n\}, M$ Fall aus SUM

$R(A, M) = A' = A \cup \{a_{n+1}, a_{n+2}\}$

mit $a_{n+1} = M + 1 \quad , \quad a_{n+2} = \sum\limits_{i=1}^{n} a_i + 1 - M$

PARTITION-OPT

Beispiel

$A = \{5, 5, 3, 3, 2\}$

$D = \{5, 5\}$

$\mu(D) = 5 + 5 = 10$

$\mu(A \setminus D) = 3 + 3 + 2 = 8$

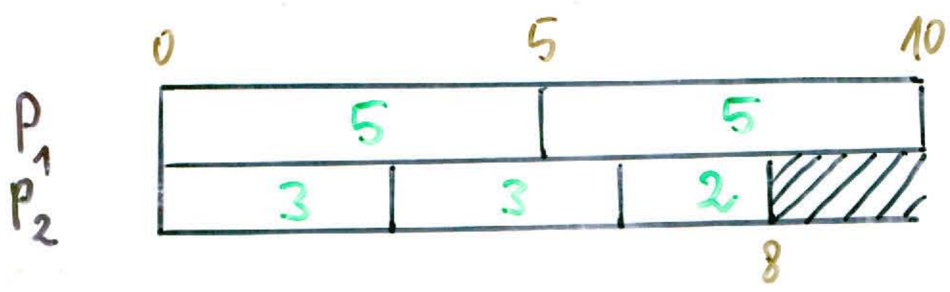
$W_{opt}(A) = 10$

Andere Interpretation: 2-SCHEDULING

a_i Bearbeitungszeit für Aufgabe T_i

2 Prozessoren

Finde Aufgabenzuteilung auf Prozessoren derart, dass Gesamtbearbeitungszeit minimal wird.



Listenschedule (Graham)

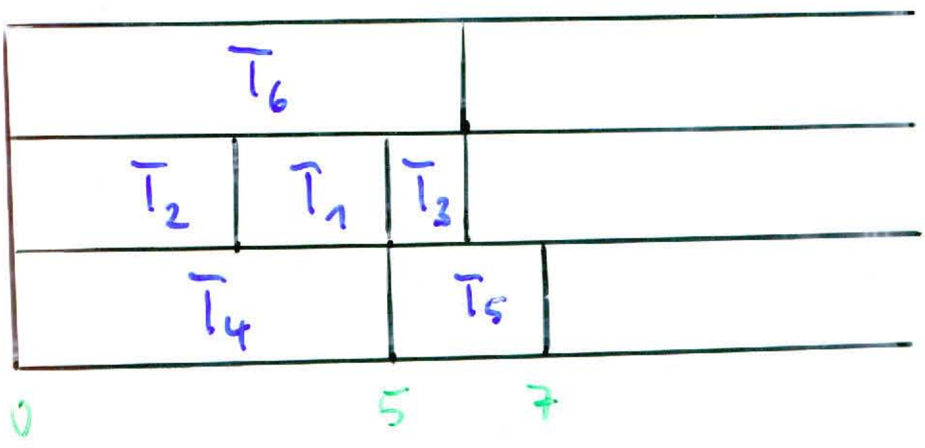
- m Maschinen $M_1, \dots, M_m$
  alle Maschinen gleich schnell
- n Aufgaben (Jobs) in Liste
  $(T_1, T_2, \dots, T_n)$ gegeben

- Listenscheduling
  zu jedem Zeitpunkt, an dem eine Maschine frei ist oder wird, ordne die nächste (verfügbare) Aufgabe aus der Liste dieser Maschine zu

Listenschedule

T_1	T_2	T_3	T_4	T_5	T_6	Aufgabe
2	3	1	5	2	6	Bearbeitungszeit (Processing Time)

- gutes Ergebnis für Liste
  $(T_6, T_2, T_4, T_1, T_3, T_5)$
  $(6, 3, 5, 2, 1, 2)$



- optimal, denn
  $\sum_{i=1}^{6} p_i = 19$ aufsummierte Bearbeitungszeiten für alle Aufgaben

  $m = 3$ Maschinen $\Rightarrow$ mindestens eine Maschine braucht $19/3 = 6\,1/3$
  $\Rightarrow \lceil 6\,1/3 \rceil = 7$ wegen Ganzzahligkeit