Pseudopolynomielle Algorithmen

Stefan Mittrich
06. April 2010

Inhalt

- Einführung
  - Inhalt
  - Motivation
- Pseudopolynomielle Algorithmen
  - Knapsackproblem
- Starke *NP*-Vollständigkeit
  - TSP

Motivation

- Frage zu Primzahlberechnung:
  - Ein einfacher Test ob n Primzahl?

Motivation

- Frage zu Primzahlberechnung:
  - Ein einfacher Test ob n Primzahl?
    - Probedivision
    - *n* durch Primzahlen zwischen n und $\sqrt{n}$ teilbar?

Motivation

- Primzahlbeispiel zeigt
  - Auch (NP) schwere Probleme müssen im Alltag gelöst werden

- Idee:
  - Schweres Problem durch Einschränkung in effizient lösbares Problem wandelbar?
    - Pseudopolynomielle Algorithmen – Beschänkung der Eingabelänge
    - Approximationsalgorithmen

Definition: Knapsackproblem

- Eingaben
  - Gewichte $g_1, g_2, \ldots g_n \in \mathbb{N}$
  - Nutzenwert $a_1, a_2, \ldots a_n \in \mathbb{N}$
  - Gewichtsschranke G

- Gesucht
  - Optimale Bepackung des Rucksacks
  - mit Einschränkung: Betrachtung Teilproblem $KP(k,g)$

Dynamische Programmierung

- Teilproblem $KP(k,g)$
  - $1 \leq k \leq n \qquad k \in \mathbb{N}$
  - $0 \leq g \leq G$
  - $N(k,g)$ *ist damit der Nutzwert der optimalen Lösung*
  - *Randwerte:*

$$\boxed{\begin{array}{ll} N(k, g) \coloneqq -\infty & \text{für } g < 0 \\ N(0, g) \coloneqq N(k, 0) \coloneqq 0 & \text{für } g \geq 0 \end{array}}$$

Iterative Lösungen

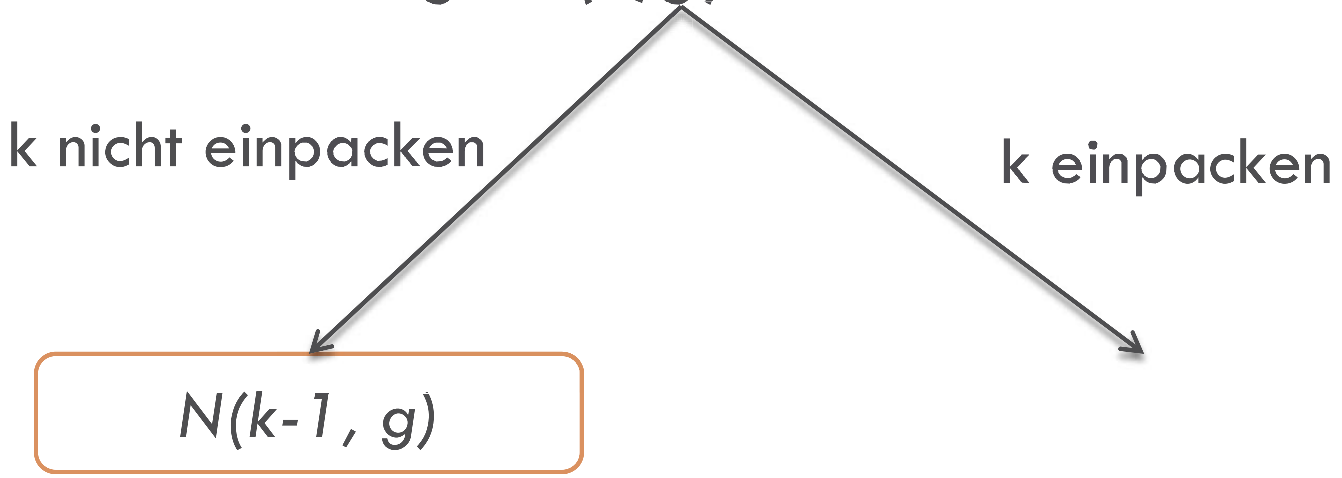
- Betrachtung $KP(k,g)$ für Element k

k nicht einpacken

k einpacken

Iterative Lösungen

- Betrachtung KP(k,g) für Element k



Iterative Lösungen

- Betrachtung KP(k,g) für Element k

k nicht einpacken

k einpacken

$N(k\text{-}1,\ g)$

$N(k\text{-}1,\ g\text{-}g_k) + a_k$

Iterative Lösungen

- Betrachtung KP(k,g) für Element k

k nicht einpacken

k einpacken

$N(k\text{-}1,\ g)$

$N(k\text{-}1,\ g\text{-}g_k) + a_k$

$$\boxed{N(k,g) = max\ \{\ N(k\text{-}1,\ g),\ N(k\text{-}1,\ g\text{-}g_k) + a_k\ \}}$$

Beispiel

- konkretes Beispiel:

k	g_k [Gewicht]	a_k [Nutzen]
1	30	100
2	40	150
3	50	120
4	20	80
5	10	50

- Ziel: Bestimme optimale Lösung mit $G = 100$

Beispiel

- Initialisierung

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1										
2										
3										
4										
5										

$$N(k,g) = \max \{ N(k-1, g), N(k-1, g-g_k) + a_k \}$$

Beispiel

- Objekt $k = 1$ $g_1=30$ $a_1=100$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0									
2										
3										
4										
5										

$$N(1,10) = \max \{ \boldsymbol{N(0, 10)}, N(0, 10\text{-}30) + 100 \}$$

Beispiel

- Objekt $k = 1$ $g_1 = 30$ $a_1 = 100$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0								
2										
3										
4										
5										

$N(1,20) = \max \{ \mathbf{N(0, 20)}, N(0, 20\text{-}30) + 100 \}$

Beispiel

- Objekt $k = 1$ $g_1=30$ $a_1=100$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0	100							
2										
3										
4										
5										

$$N(1,30) = \max \{N(0, 30), \boldsymbol{N(0, 30\text{-}30) + 100}\}$$

Beispiel

- Objekt $k = 1$ $g_1=30$ $a_1=100$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0	100	100						
2										
3										
4										
5										

$N(1{,}40) = \max\{N(0, 40), \boldsymbol{N(0, 40\text{-}30) + 100}\}$

Beispiel

- Objekt $k = 1$ $g_1 = 30$ $a_1 = 100$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0	100	100	100	100	100	100	100	100
2										
3										
4										
5										

$$N(1,100) = \max\{\, N(0, 100), \mathbf{N(0, 100\text{-}30) + 100} \,\}$$

Beispiel

- Objekt $k = 2$ $g_2=40$ $a_2=150$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0	100	100	100	100	100	100	100	100
2	0	0	100							
3										
4										
5										

$N(2,30) = \max \{ \boldsymbol{N(1, 30)}, N(1, 30\text{-}40) + 150 \}$

Beispiel

- Objekt $k = 2$ $g_2=40$ $a_2=150$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0	100	100	100	100	100	100	100	100
2	0	0	100	150						
3										
4										
5										

$$N(2{,}40) = \max \{ N(1, 40), \boldsymbol{N(1, 40\text{-}40) + 150} \}$$

Beispiel

- Objekt $k = 2$ $g_2=40$ $a_2=150$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0	100	100	100	100	100	100	100	100
2	0	0	100	150	150	150	250			
3										
4										
5										

$$N(2,70) = \max \{ N(1, 70), \mathbf{N(1, 70\text{-}40) + 150} \}$$

Beispiel

- Objekt $k = 5$ $g_5=10$ $a_5=50$

	10	20	30	40	50	60	70	80	90	100
0	0	0	0	0	0	0	0	0	0	0
1	0	0	100	100	100	100	100	100	100	100
2	0	0	100	150	150	150	250	250	250	250
3	0	0	100	150	150	150	250	250	270	270
4	0	80	100	150	180	230	250	250	330	330
5	50	80	130	150	180	230	280	300	330	380

$N(5,100) = \max \{ N(4, 100), \mathbf{N(4, 100\text{-}10) + 50}\}$

Ergebnis

- Fazit:
  - Tabelle mit n Zeilen und G Spalten
  - Jeder Tabelleneintrag $O(1)$
  - → Gesamte Tabelle $O(nG)$

Satz 1:
Knapsackproblem in Zeit $O(nG)$ lösbar

P = NP ?

- Betrachtung der Eingabelänge

$$\text{Eingabe}: \; n, g_1, g_2, \ldots, g_n, G, a_1, \ldots, a_n \in \{0, 1\}^*$$

  - mit der Annahme

$$g_i \leq G \quad \text{für } i \in \{0 \ldots k\}$$
$$a_{max} = \max \{ a_1, \ldots, a_n \}$$

- Eingabelänge beschränkt durch

$$O(n (\log_2 G + \log_2 a_{max}))$$