



openHPI
Einführung in die Mathematik der Algorithmik
– Dijkstra Analyse II

Pascal Lenzner und Karen Seidel

Hasso-Plattner-Institut

Universität Potsdam

G ist zusammenhängend

Betrachten Graph G mit n Knoten und
 $m \geq n - 1$ Kanten

Gesamtkosten:

$$\begin{aligned} &\Theta(n \cdot \text{cost}(\text{Insert})) \\ &+ \Theta(n \cdot \text{cost}(\text{ExtractMin})) \\ &+ \Theta(m \cdot \text{cost}(\text{DecreaseKey})) \\ &+ \Theta(n) + \Theta(m) \end{aligned}$$

Kosten der Priority-Queue Operationen bei n Elementen:

	sortierte Liste	binärer Min-Heap
ExtractMin	$\Theta(1)$	$\Theta(\log_2(n))$
Insert	$\Theta(n)$	$\Theta(\log_2(n))$
DecreaseKey	$\Theta(n)$	$\Theta(\log_2(n))$

Gesamtkosten mit sortierter
Liste als PQ:

$$\Theta(m \cdot n)$$

Initialisierung:

- nur Knoten H markiert; Distanz zum Startknoten $H = 0$
- alle Nachbarn X des Startknotens H sind Randknoten mit $\text{dist}(X, H) = \text{Länge}(X, H)$; Vorgänger ist H

Solange unmarkierter Randknoten existiert:

- X sei unmarkierter Randknoten mit geringster Distanz zu H
- entferne X aus der Menge der Randknoten
- mache für alle unmarkierten Nachbarn Y von X folgendes:
 - falls Y kein Randknoten:
 - füge Y in die Menge der Randknoten ein
 - $\text{dist}(Y, H) = \text{Länge}(Y, X) + \text{dist}(X, H)$; Vorgänger von Y ist X
 - sonst:
 - falls $\text{dist}(Y, H) > \text{Länge}(Y, X) + \text{dist}(X, H)$:
 - aktualisiere $\text{dist}(Y, H)$ zu $\text{Länge}(Y, X) + \text{dist}(X, H)$
 - X wird Vorgänger von Y
- markiere X

Gesamtkosten mit binärem
Min-Heap als PQ:

$$\Theta(m \cdot \log_2(n))$$

Gesamtkosten mit sortierter
Liste als PQ:

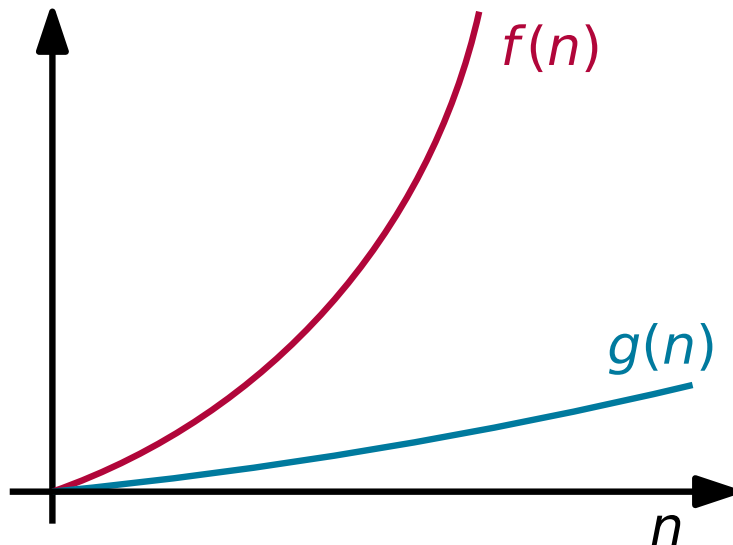
$$\Theta(m \cdot n)$$

Gesamtkosten mit binärem
Min-Heap als PQ:

$$\Theta(m \cdot \log_2(n))$$

Welche Version ist besser?

- vergleichen das asymptotische Wachstum der Funktionen $f(n) = m \cdot n$ und $g(n) = m \cdot \log_2(n)$:



Es gilt $g(n) \in o(f(n))$.

- für genügend große n ist Dijkstra mit binärem Min-Heap klar besser!
- eine passende Datenstruktur ist entscheidend für die Güte eines Algorithmus!