

Zu verwendende Ziffer *als* n

Ergebnis der Rechnung *als* m

//in $dist[x]$ wird die kleinste Anzahl der verwendeten n in einer Rechnung mit Ergebnis x gespeichert

$dist =$ *Liste* aus Zahlen

$a =$ *Menge* aller Ergebnisse für die eine Rechnung gefunden wurde

$pq =$ *Menge* aller bekannten (verwendete n , Ergebnis) Paare

füge alle n , nn , nnn , ... in pq ein

//Dijkstra-Algorithmus

solange Größe von $pq > 0$

$d, k =$ kleinstes Element in pq

lösche kleinstes Element aus pq

//wenn schon eine Rechnung mit weniger n und Ergebnis k gefunden wurde, dann wieder zum Schleifenkopf springen

wenn ($d \geq dist[k]$) *continue*

//wenn k das gesuchte Ergebnis ist, dann den Algorithmus abbrechen

wenn ($k == m$) *break*

//dist updaten

ändere $dist[k]$ d

//Rechnungen mit k in pq Eintragen

für jede Zahl $k1$ in a *mache*

trage $\{dist[k]+dist[u], k+u\}$ in pq ein

wenn ($k-u \geq 0$)

trage {*dist*[*k*]+*dist*[*u*], *k-u*} in *pq* ein
wenn (*u-k* >= 0)
trage {*dist*[*k*]+*dist*[*u*], *u-k*} in *pq* ein
trage {*dist*[*k*]+*dist*[*u*], *k*u*} in *pq* ein
wenn (*k/u* ganzzahlig)
trage {*dist*[*k*]+*dist*[*u*], *k/u*} in *pq* ein
wenn (*u/k* ganzzahlig)
trage {*dist*[*k*]+*dist*[*u*], *u/k*} in *pq* ein

weise jeder Kreuzung eine Nummer zu

//in $dist[k][a]$ wird die kürzeste Distanz zur Kreuzung k gespeichert, wenn bis zu a mal abgebogen wurde

$dist = \mathbf{Tabelle}$ mit der Größe $anz. \text{ Kreuzungen} \times anz. \text{ Kreuzungen}$

$pq = \mathbf{Menge}$ aller bekannten ($anz. \text{ Abgebogen}, \text{ Distanz}, \text{ Kreuzung}$) Tripel

füge Startkreuzung in pq ein

//Abgewandelte Form des Dijkstra-Algorithmus

solange Größe von $pq > 0$

$a, d, k =$ kleinstes Element in pq

lösche kleinstes Element aus pq

//wenn schon ein kürzerer Weg zu k gefunden wurde, dann wieder zum Schleifenkopf springen

wenn ($d \geq dist[k][a]$) **continue**

//betreffende Einträge in $dist$ ändern

zähle in i von a aufwärts

ändere $dist[k][i]$ auf d

//alle von k aus durch geradeaus-fahren erreichbaren Knoten in pq eintragen

gehe alle durch geradeaus-fahren von k erreichbaren Knoten u **durch**

$d1 =$ Distanz zwischen k und u

trage $\{a+1, d+d1, u\}$ in pq ein