Aufgabe 1

Lemma: Gegeben ist ein Graph $G_c = (V, E)$ mit Kosten $c(E)$. Für Mengen A, B mit transitiven, totalen und antisymmetrischen Ordnungsrelationen $<_A, <_B$ und eine Funktion $f : A \to B$ gilt die folgende Äquivalenz:

$$f \text{ erhält die Relationen auf } A, B \Leftrightarrow \text{ein MST von } G_c \text{ ist auch MST von } G_{f(c)}$$

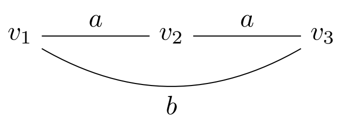
Beweis. Richtung $\Rightarrow$:

f erhält die Ordnung auf den Kantengewichten. Insbesondere ist somit die kürzeste Kante aus G_c identisch mit der kürzesten Kante aus $G_{f(c)}$. Wenn der Algorithmus von Kruskal daher $e \in E$ wählt, kann er auf dem geänderten Graphen sie selbe Kante e wählen. Iterativ kann der Algorithmus so den selben MST aufbauen. Da der Algorithmus von Kruskal korrekt ist, ist der auf $G_{f(c)}$ berechnete Spannbaum identisch zum MST auf G_c und minimal, also ein MST.

Richtung $\Leftarrow$:

Zu zeigen: erhält f die Ordnung nicht, so existiert ein Graph H_c, für den ein MST existiert, der auf $H_{f(c)}$ kein MST ist.

Sei also f eine Funktion, die die Ordnung nicht erhält. Es existieren also zwei Elemente a, b, so dass $a <_A b$, aber $f(b) <_B f(a)$. Betrachte nun den folgenden Graphen:



Auf H_c ist ein MST mit Gewicht $2 \cdot a$ durch die Kanten $\{(v_1, v_2), (v_2, v_3)\}$ gegeben. Auf $H_{f(c)}$ ist dieser Spannbaum mit Gewicht $2 \cdot f(a)$ jedoch kein MST, da der durch die Kanten $\{(v_1, v_2), (v_1, v_3)\}$ gegebene Baum mit Gewicht $f(a) + f(b) <_Y 2 \cdot f(a)$ bereits ein kleineres Gesamtgewicht hat. $\square$

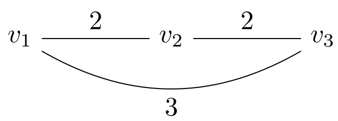
Die Aufgabe (a) folgt als Korollar aus diesem Lemma.

(a)

Die Funktion $f : x \mapsto x^2$ ist auf den positiven, reellen Zahlen streng monoton und erhält somit die Ordnungsrelation $<$. Die Aussage folgt somit aus obigem Lemma.

(b)

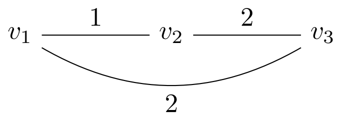
Das folgende Beispiel mit $X = \{v_2\}$ widerlegt die Behauptung.



Der MST $\{(v_1, v_2), (v_2, v_3)\}$ ist auf dem geänderten Graphen kein MST.

(c)

Das folgende Beispiel mit $X = \{v_1, v_2\}$ widerlegt die Behauptung.



Der MST $\{(v_1, v_2), (v_2, v_3)\}$ ist auf dem geänderten Graphen kein MST.

Aufgabe 2

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
init	0	∞	∞	∞	∞	∞	∞	∞
$x_1 \xrightarrow{1} x_8$	0	∞	∞	∞	∞	∞	∞	1
$x_1 \xrightarrow{2} x_2$	0	2	∞	∞	∞	∞	∞	1
$x_1 \xrightarrow{2} x_7$	0	2	∞	∞	∞	∞	2	1
$x_1 \xrightarrow{3} x_5$	0	2	∞	∞	3	∞	2	1
$x_2 \xrightarrow{1} x_3$	0	2	3	∞	3	∞	2	1
$x_8 \xrightarrow{2} x_6$	0	2	3	∞	3	3	2	1
$x_8 \xrightarrow{4} x_4$	0	2	3	5	3	3	2	1

Aufgabe 3

Aufgabe 4

(a)

Der Algorithmus berechnet kürzeste Wege bezüglich c', da Dijktras Algorithmus mit den Gewichten c' ausgeführt wird.

Behauptung: Hat ein Weg von s nach t über c eine Gesamtlänge von w, so hat er über c' ebenfalls die Gesamtlänge von w'.

Beweis: Die Gesamtlänge über c berechnet sich durch $w = c(s, v_1) + c(v_1, v_2) + ... + c(v_n, t)$. Über c' ergibt sich $w' = c(s, v_1) + d(v_1, t) - d(s, t) + c(v_1, v_2) + d(v_2, t) - d(v_1, t) + ... + c(v_n, s) + d(t, t) - d(v_n, t)$. Da sich die Summanden $d(v_i, t)$ (ähnlich einer Teleskopsumme) auslöschen, verbleibt $w' = c(s, v_1) - d(s, t) + c(v_1, v_2) + ... + c(v_n, s) + d(t, t) = c(s, v_1) + c(v_1, v_2) + ... + c(v_n, t) - d(s, t) = w$.

Die Gesamtlänge der Pfade ändert sich also nur um die konstante Länge $d(s, t)$, Dijkstra liefert also (analog zum Lemma aus Aufgabe 1) über c' als kürzesten Pfad einen Pfad, der so lang ist wie der kürzeste über c. Der Algorithmus berechnet daher auch bezüglich c kürzeste Wege.

Der angepasste Algorithmus bevorzugt durch die veränderten Gewichte Kanten, die in Richtung des Ziels t führen. Dies entspricht im wesentlichen der Idee von A^*, dadurch soll der kürzeste Weg schneller gefunden und somit weniger Aktualisierungen der Entfernungen durchgeführt werden.

(b)