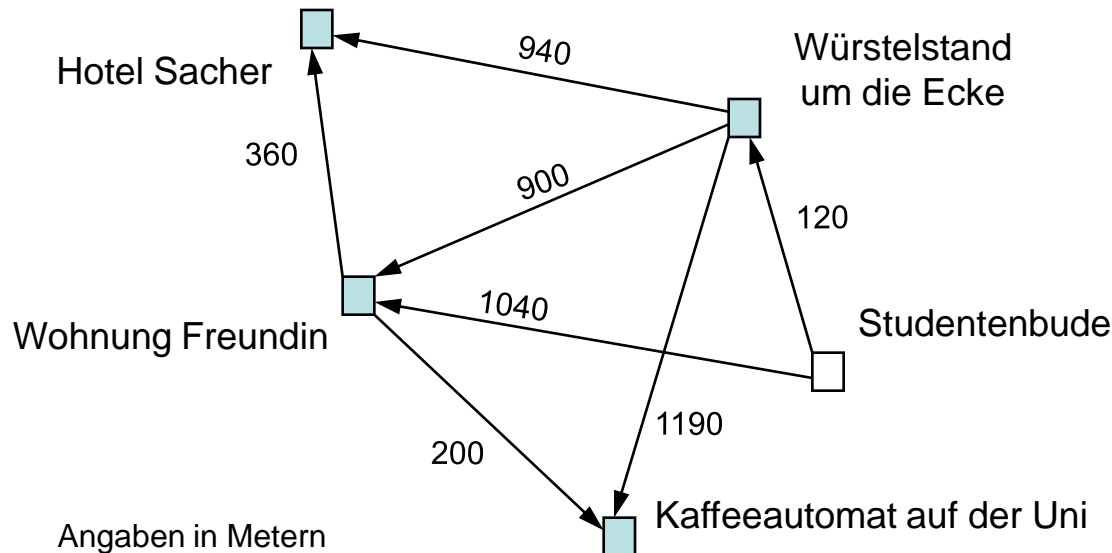


## Ablauf Dijkstra Algorithmus an unserem Beispiel:

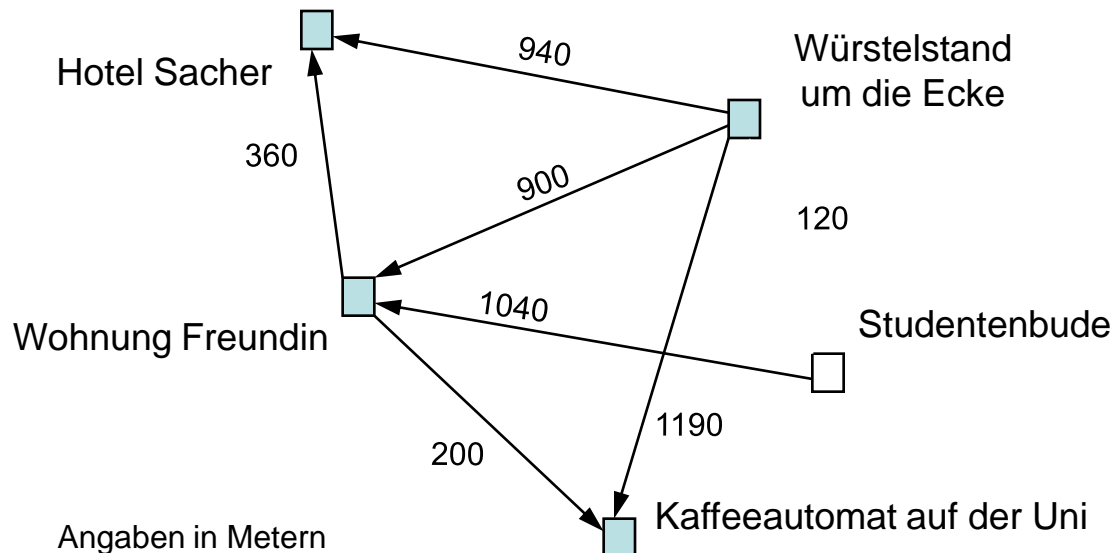
	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	$\infty$	$\infty$	$\infty$	$\infty$
Bearbeitet?	ja	nein	nein	nein	nein
Vorgänger-knoten	-	-	-	-	-



Prozedur Dijkstra (A);  
 Initialisiere  
     Distanz =  $\infty$  ,  
     Vorgängerknoten = leer,  
     bearbeitet = „nein“  
 für alle Knoten;  
 Setze  
     bearbeitet auf „ja“ und Distanz = 0  
 für den Startknoten A; /\* A = Studentenbude \*/

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	$\infty$	$\infty$	$\infty$
Bearbeitet?	ja	nein	nein	nein	nein
Vorgänger-knoten	-	Studentenbude	-	-	-



Sei x der zuletzt auf bearbeitet = „ja“ markierte Knoten

*/\* x = Studentenbude \*/*

Modifiziere alle Nachbarn y von x durch

*/\* y = Würstelstand um die Ecke \*/*

Falls  $Distanz(y) > (Distanz(x) + Edge(x,y))$  dann

*/\*  $\infty > (0 + 120)$  \*/*

$Distanz(y) := (Distanz(x) + Edge(x,y));$

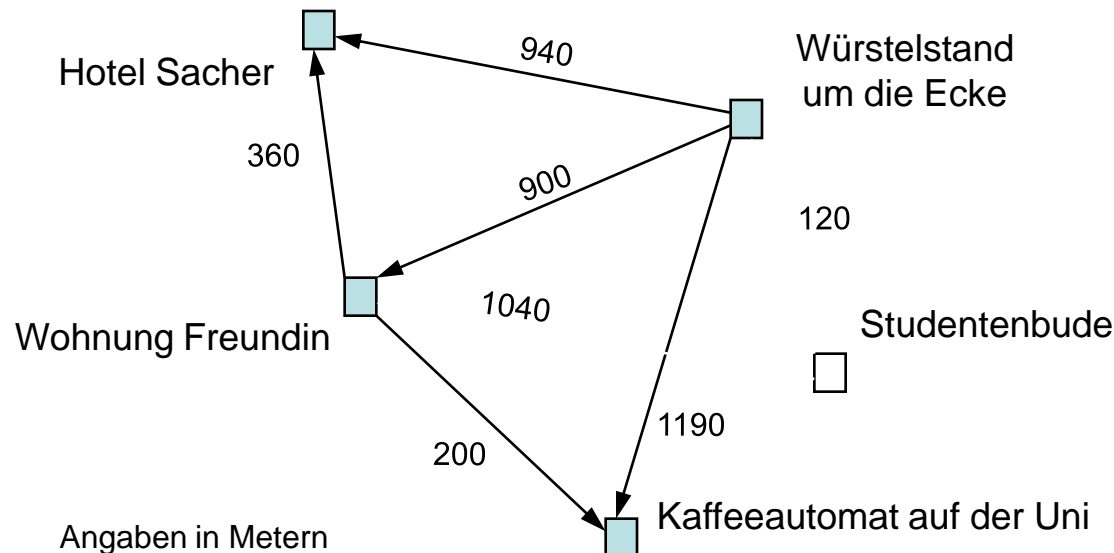
*/\* Distanz(y) = 120; \*/*

$Vorgänger(y) := x;$

*/\* Vorgänger(y) = Studentenbude; \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	$\infty$	1040	$\infty$
Bearbeitet?	ja	nein	nein	nein	nein
Vorgänger-knoten	-	Studentenbude	-	Studentenbude	-



Sei x der zuletzt auf bearbeitet = „ja“ markierte Knoten

*/\* x = Studentenbude \*/*

Modifiziere alle Nachbarn y von x durch

*/\* y = Wohnung Freundin \*/*

Falls  $Distanz(y) > (Distanz(x) + Edge(x,y))$  dann

*/\*  $\infty > (0 + 1040)$  \*/*

$Distanz(y) := (Distanz(x) + Edge(x,y));$

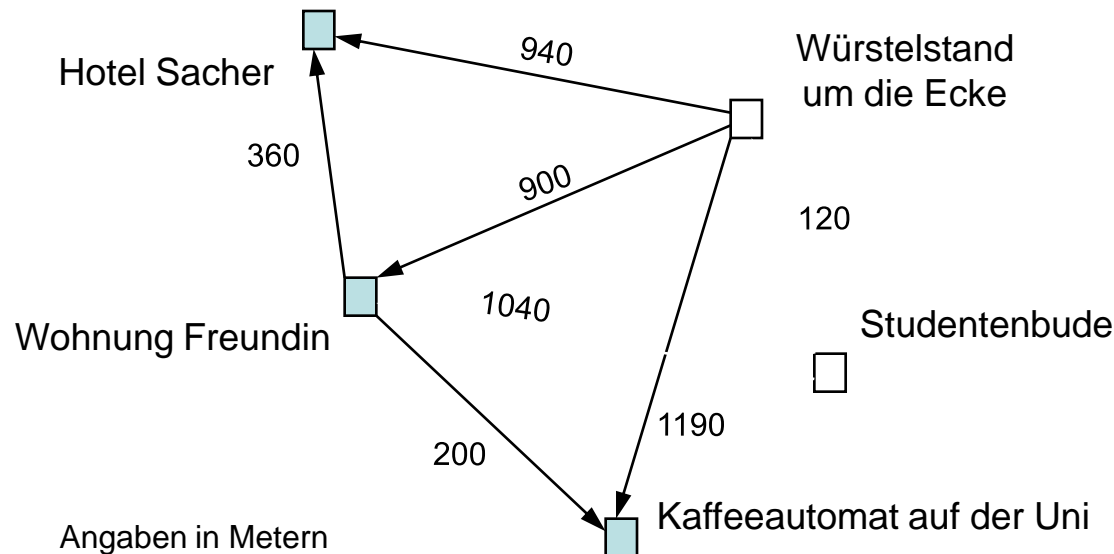
*/\* Distanz(y) = 1040; \*/*

$Vorgänger(y) := x;$

*/\* Vorgänger(y) = Studentenbude; \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

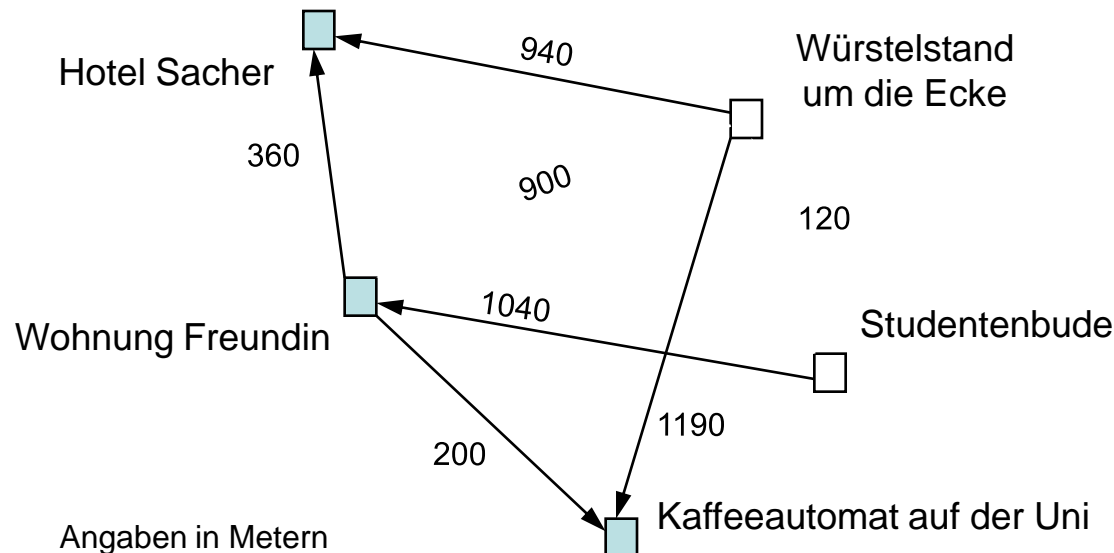
	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	$\infty$	1040	$\infty$
Bearbeitet?	ja	ja	nein	nein	nein
Vorgänger-knoten	-	Studentenbude	-	Studentenbude	-



Setze für den Knoten mit der kleinsten vorläufigen Distanz das Kennzeichen bearbeitet auf „ja“;  
*/\* bearbeitet(Würstelstand um die Ecke) = ja \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	$\infty$	1020	$\infty$
Bearbeitet?	ja	ja	nein	nein	nein
Vorgänger-knoten	-	Studentenbude	-	Würstelstand um die Ecke	-



Sei x der zuletzt auf bearbeitet = „ja“ markierte Knoten;

*/\* x = Würstelstand um die Ecke \*/*

Modifiziere alle Nachbarn y von x durch

*/\* y = Wohnung Freundin \*/*

Falls  $Distanz(y) > (Distanz(x) + Edge(x,y))$  dann

*/\*  $1040 > (120 + 900)$  \*/*

$Distanz(y) := (Distanz(x) + Edge(x,y));$

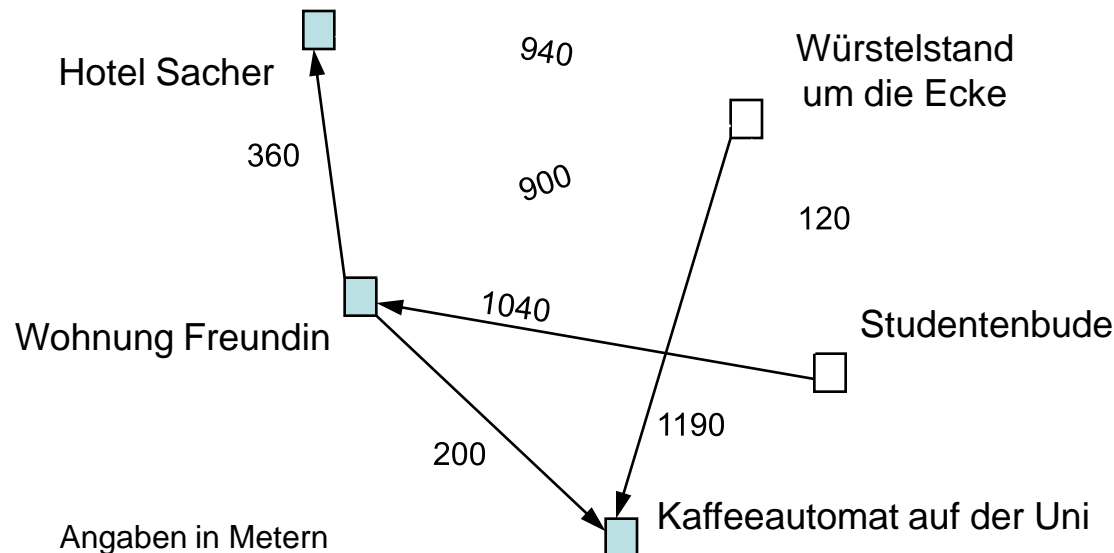
*/\*  $Distanz(y) = 1020;$  \*/*

$Vorgänger(y) := x;$

*/\*  $Vorgänger(y) = Würstelstand um die Ecke;$  \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	$\infty$
Bearbeitet?	ja	ja	nein	nein	nein
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	-



Sei x der zuletzt auf bearbeitet = „ja“ markierte Knoten;

*/\* x = Würstelstand um die Ecke \*/*

Modifiziere alle Nachbarn y von x durch

*/\* y = Hotel Sacher \*/*

Falls  $Distanz(y) > (Distanz(x) + Edge(x,y))$  dann

*/\*  $\infty > (120 + 940)$  \*/*

$Distanz(y) := (Distanz(x) + Edge(x,y));$

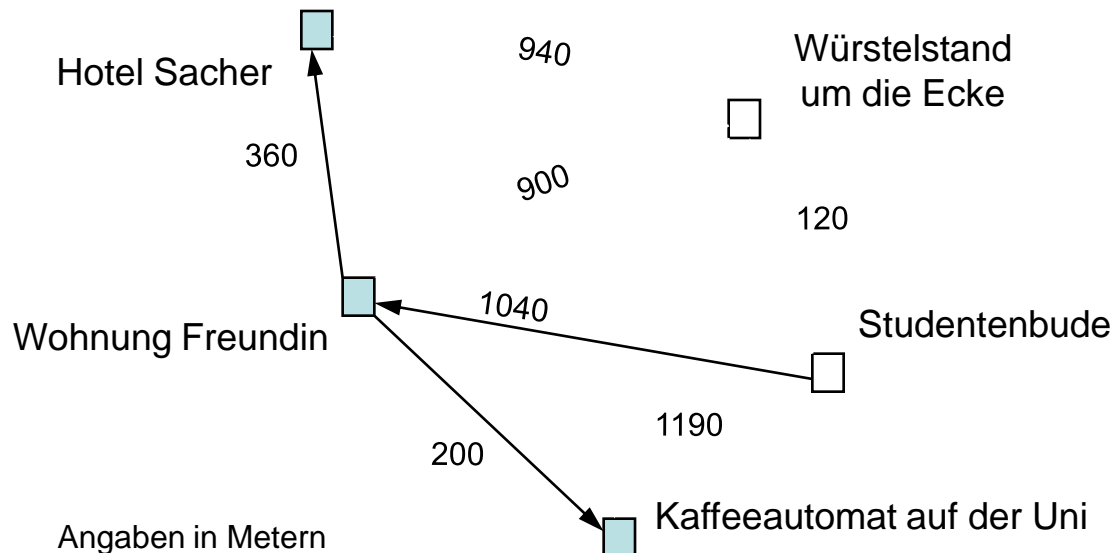
*/\* Distanz(y) = 1060; \*/*

$Vorgänger(y) := x;$

*/\* Vorgänger(y) = Würstelstand um die Ecke; \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat Uni
Distanz	0	120	1060	1020	1310
Bearbeitet?	ja	ja	nein	nein	nein
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Würstelstand um die Ecke



Sei x der zuletzt auf bearbeitet = „ja“ markierte Knoten;

*/\* x = Würstelstand um die Ecke \*/*

Modifiziere alle Nachbarn y von x durch

*/\* y = Kaffeautomat auf der Uni \*/*

Falls  $Distanz(y) > (Distanz(x) + Edge(x,y))$  dann

*/\*  $\infty > (120 + 1190)$  \*/*

$Distanz(y) := (Distanz(x) + Edge(x,y));$

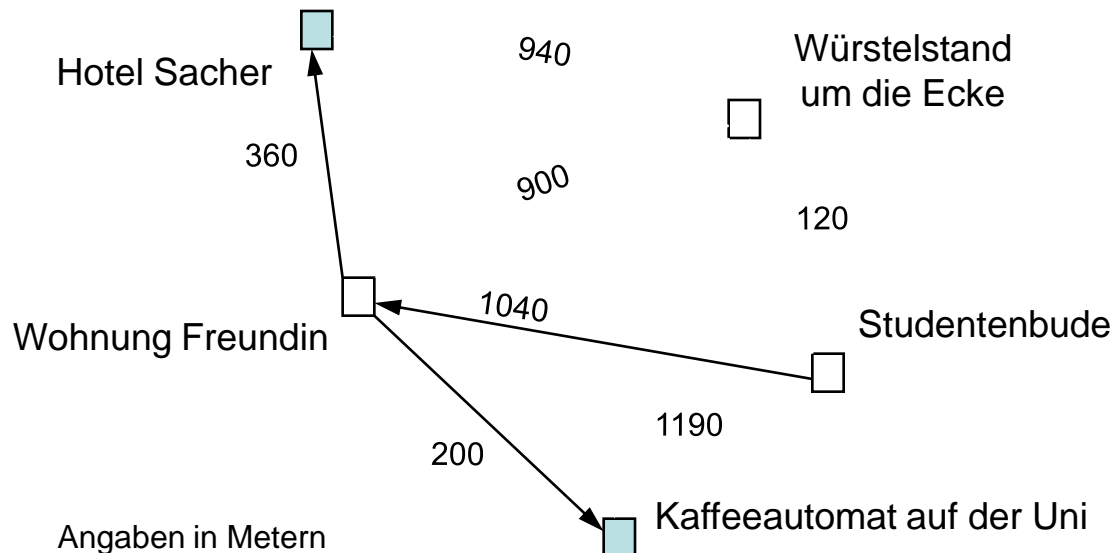
*/\* Distanz(y) = 1310; \*/*

$Vorgänger(y) := x;$

*/\* Vorgänger(y) = Würstelstand um die Ecke; \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1310
Bearbeitet?	ja	ja	nein	ja	nein
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Würstelstand um die Ecke

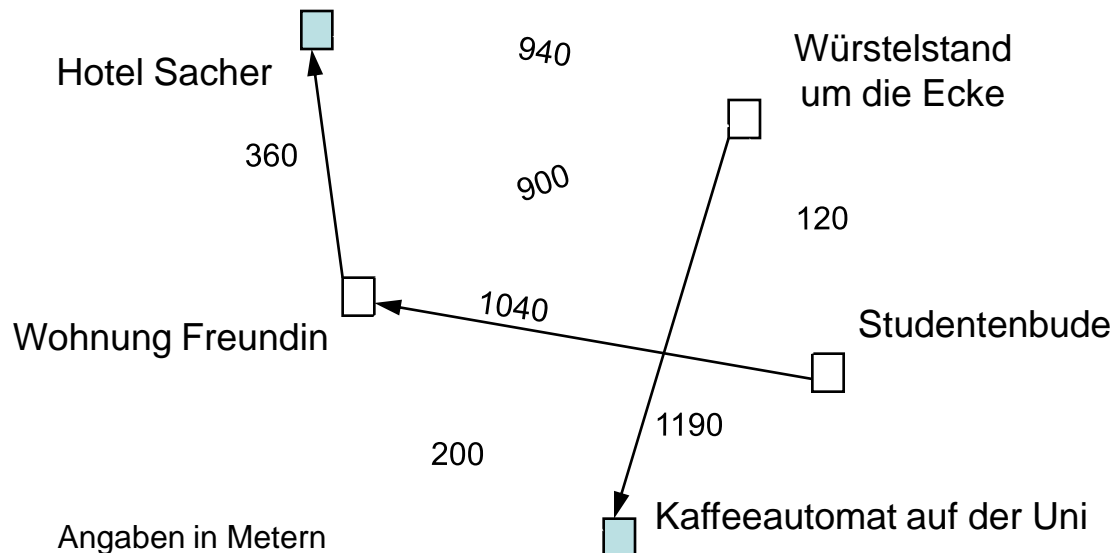


Setze für den Knoten mit der kleinsten vorläufigen Distanz das Kennzeichen bearbeitet auf „ja“;  
/\* bearbeitet(Wohnung Freundin) = ja \*/



## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	nein	ja	nein
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Sei x der zuletzt auf bearbeitet = „ja“ markierte Knoten

*/\* x = Wohnung Freundin \*/*

Modifiziere alle Nachbarn y von x durch

*/\* y = Kaffeautomat auf der Uni \*/*

Falls  $Distanz(y) > (Distanz(x) + Edge(x,y))$  dann

*/\*  $1310 > (1020 + 200)$  \*/*

$Distanz(y) := (Distanz(x) + Edge(x,y));$

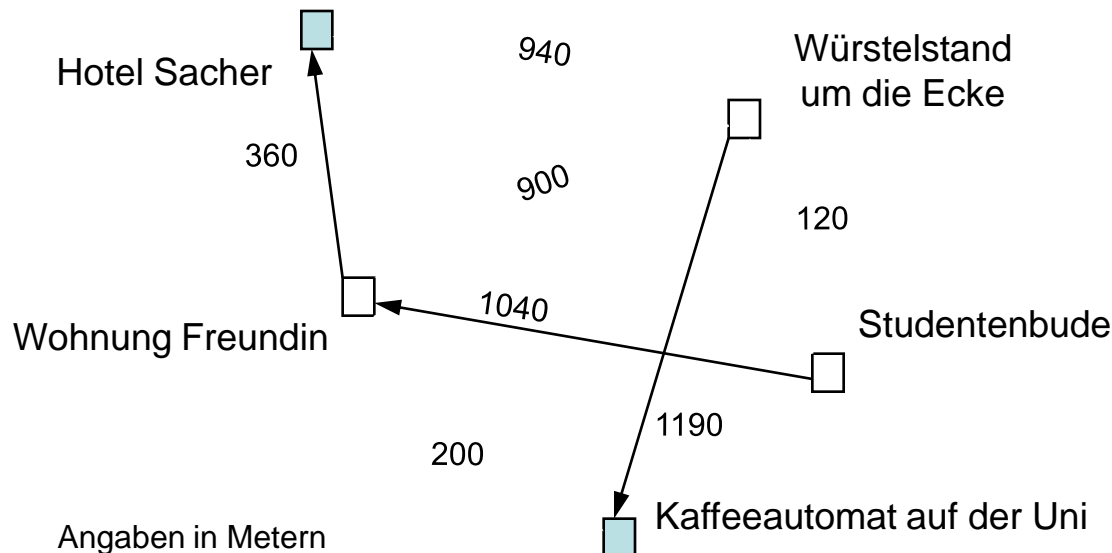
*/\*  $Distanz(y) = 1220;$  \*/*

$Vorgänger(y) := x;$

*/\*  $Vorgänger(y) = Wohnung Freundin;$  \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	nein	ja	nein
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Sei x der zuletzt auf bearbeitet = „ja“ markierte Knoten

*/\* x = Wohnung Freundin \*/*

Modifiziere alle Nachbarn y von x durch

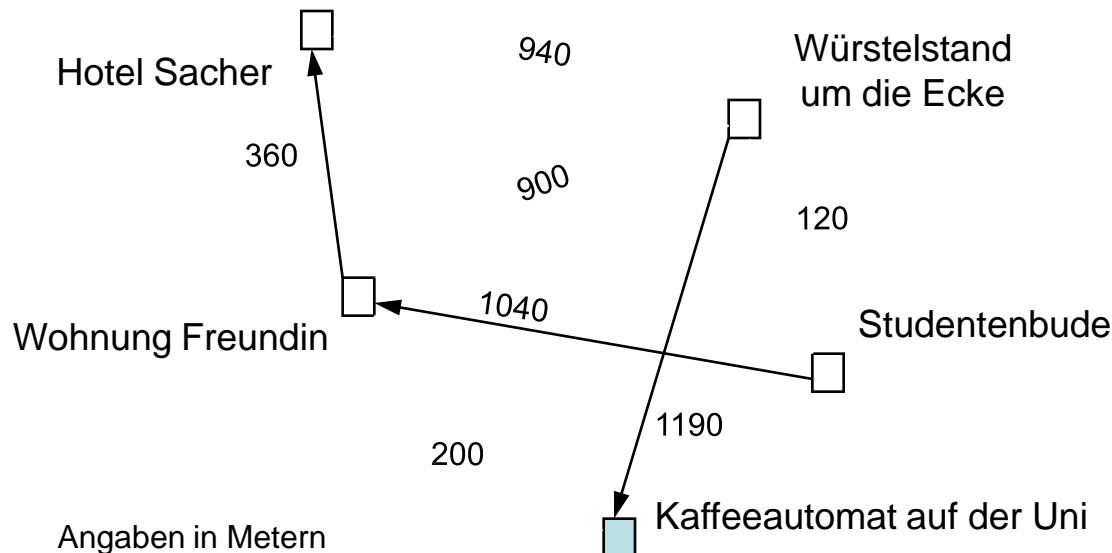
*/\* y = Hotel Sacher \*/*

Falls  $Distanz(y) > (Distanz(x) + Edge(x,y))$  dann

*/\*  $1060 > (1020 + 360)$  f.A. Keine Änderung \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

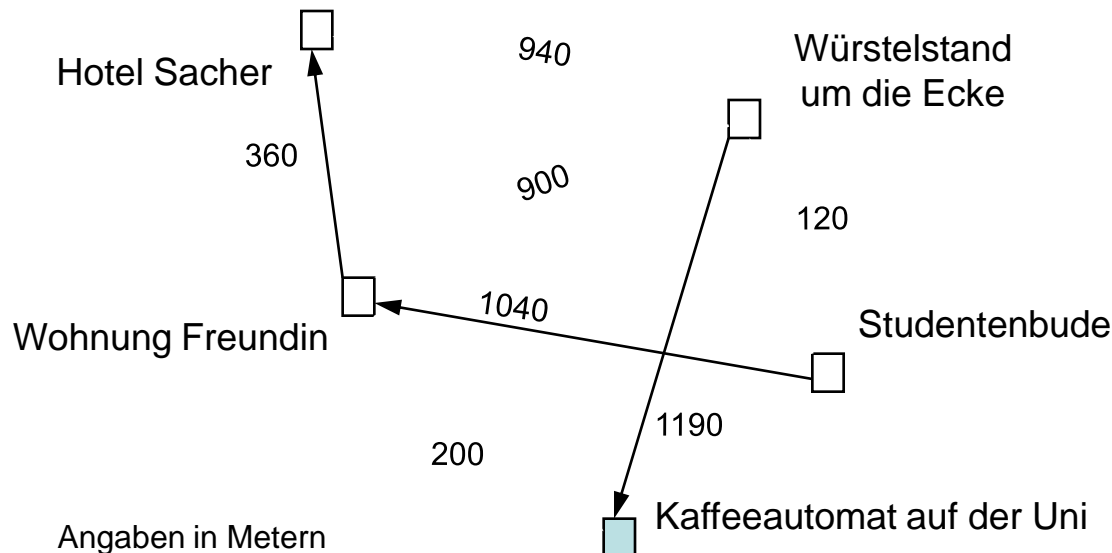
	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	nein
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Setze für den Knoten mit der kleinsten vorläufigen Distanz das Kennzeichen *bearbeitet* auf „ja“;  
*/\* bearbeitet(Hotel Sacher) = ja \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	nein
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Sei x der zuletzt auf bearbeitet = „ja“  
markierte Knoten

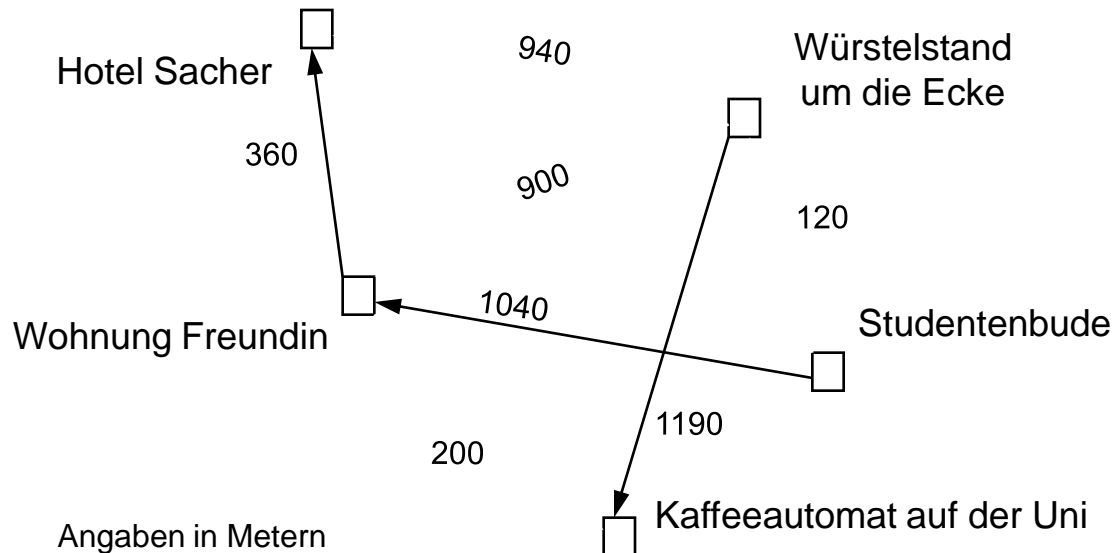
*/\* x = Hotel Sacher \*/*

Modifiziere alle Nachbarn y von x durch

*/\* Hotel Sacher hat keine Nachbarn  
nichts zu modifizieren! \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

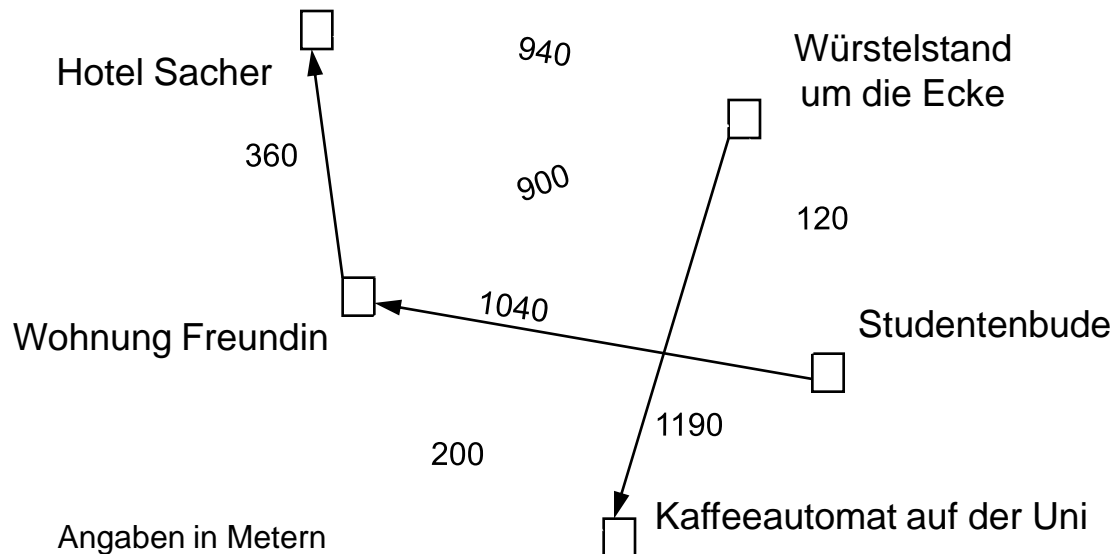
	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	ja
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Setze für den Knoten mit der kleinsten vorläufigen Distanz das Kennzeichen bearbeitet auf „ja“;  
/\* bearbeitet(Kaffeautomat auf der Uni) = ja \*/

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	ja
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Sei x der zuletzt auf bearbeitet = „ja“  
markierte Knoten

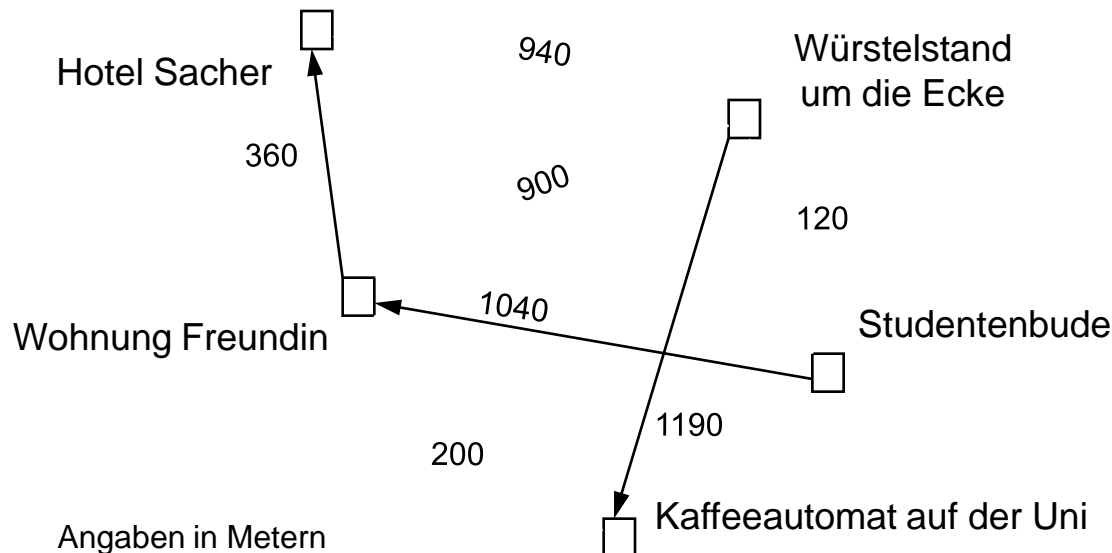
*/\* x = Kaffeautomat auf der Uni \*/*

Modifiziere alle Nachbarn y von x durch

*/\* Kaffeautomat auf der Uni hat keine Nachbarn  
nichts zu modifizieren! \*/*

## Ablauf Dijkstra Algorithmus an unserem Beispiel:

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	ja
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Repeat

·  
·  
·

Until alle Knoten bearbeitet = „ja“

/\* Algorithmus terminiert und alle kürzesten Wege von der Studentenbude aus zum nächsten wohl verdienten Kaffee sind nun gefunden! \*/

Wenn der Algorithmus durchgelaufen ist, ergibt sich die folgende Darstellung der kürzesten Wege von der Studentenbude zu allen möglichen Kaffequellen. Da der Kaffee bei seiner Freundin einfach am besten (und vor allem am billigsten!!) ist, sucht er den kürzesten Weg von seiner Studentenbude zu der Wohnung seiner Freundin.

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	ja
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin



Dazu geht er die Liste vom Zielort (Wohnung Freundin) bis zu seiner Studentenbude durch. Da immer die Vorgängerknoten gespeichert sind erhält er somit den kürzesten Weg zu seiner Freundin!

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	ja
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin

Weg: Wohnung Freundin

Der Vorgängerknoten von „Wohnung Freundin“ ist „Würstelstand um die Ecke“. Also ist das der nächste Zwischenstop auf dem kürzesten Weg zum billigsten Kaffee.

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	ja
Vorgänger-knoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin

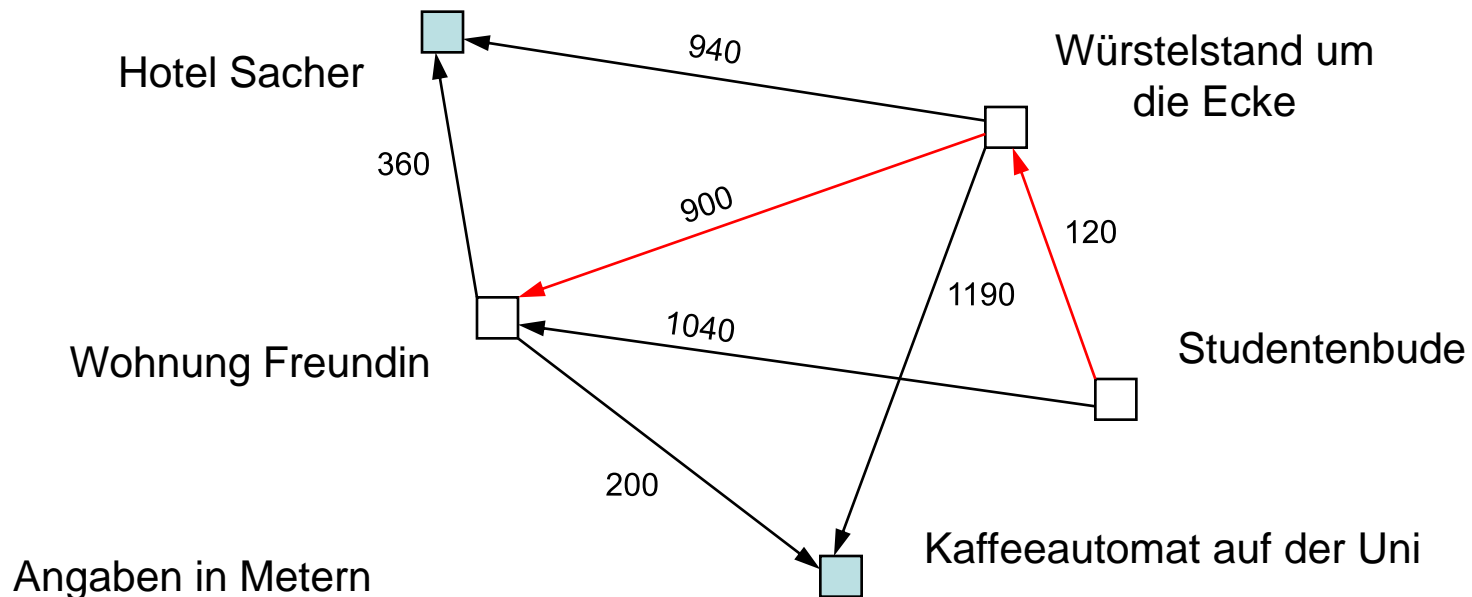
Weg: Würstelstand um die Ecke → Wohnung Freundin

Der Vorgängerknoten vom „Würstelstand um die Ecke“ ist seine „Studentenbude“. Da das der Startknoten ist, hat er nun den kürzesten Weg zur Wohnung seiner Freundin gefunden!

	Studentenbude	Würstelstand um die Ecke	Hotel Sacher	Wohnung Freundin	Kaffeautomat auf der Uni
Distanz	0	120	1060	1020	1220
Bearbeitet?	ja	ja	ja	ja	ja
Vorgängerknoten	-	Studentenbude	Würstelstand um die Ecke	Würstelstand um die Ecke	Wohnung Freundin

Weg: Studentenbude → Würstelstand um die Ecke → Wohnung Freundin

Zur Veranschaulichung noch einmal der Graph mit dem kürzesten Weg vom Knoten „Studentenbude“ zum Knoten „Wohnung Freundin“:



Hier kann man auch erkennen, daß der direkte Weg von der Studentenbude zur Wohnung der Freundin länger ist (um 20 Meter).