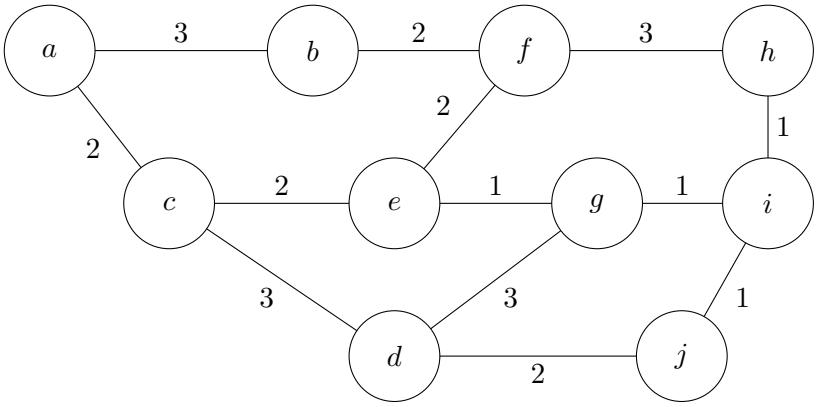
Katrin Bott (katrin.bott@student.uni-tuebingen.de, 3944875)
Mila Gorecki (mila.gorecki@student.uni-tuebingen.de, 3962077)

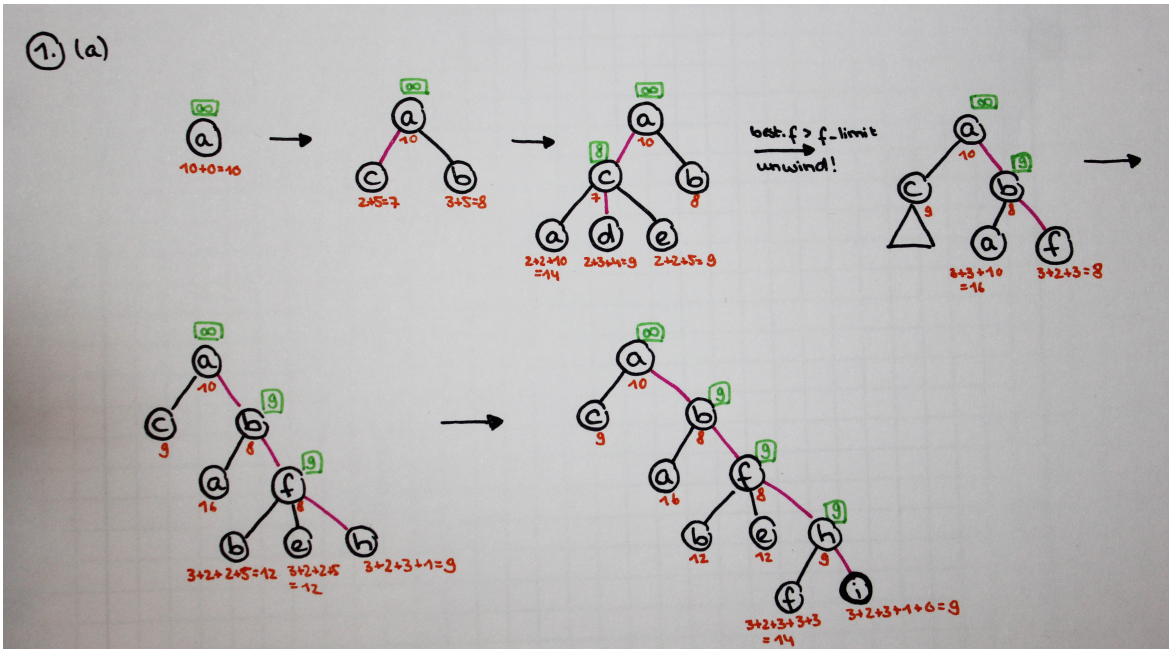
Assignment Nr. 4

(Abgabetermin 23.11.2016)

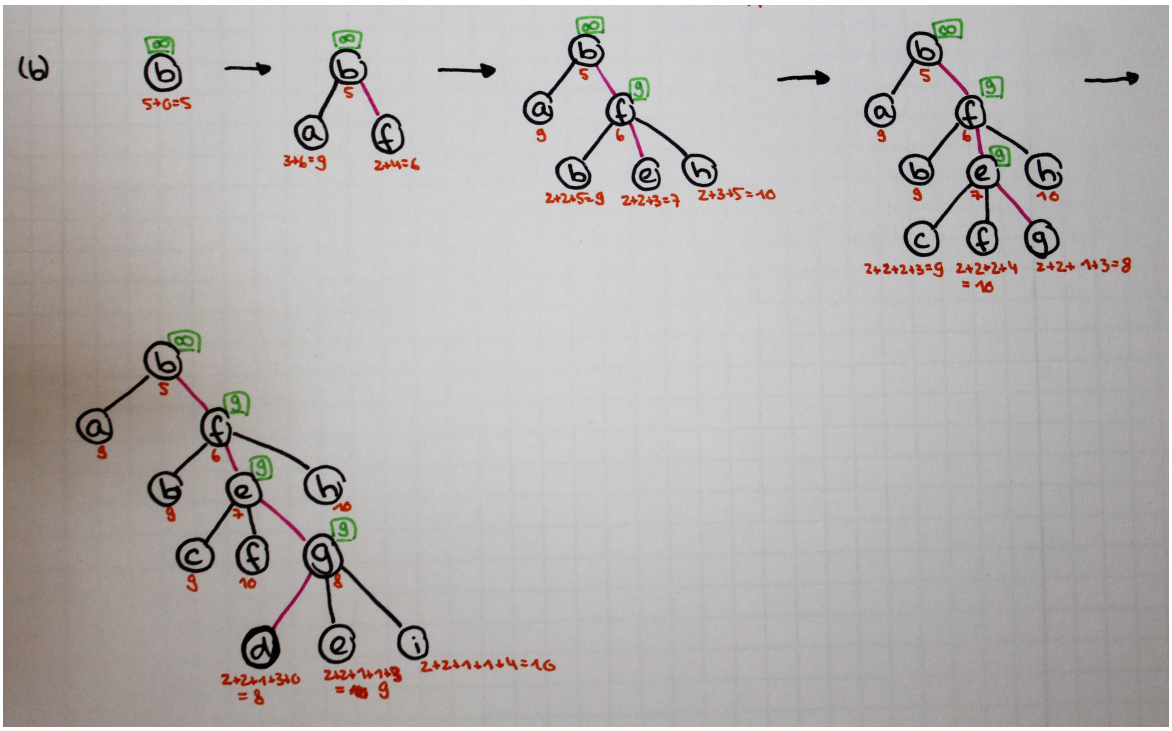
Question 1



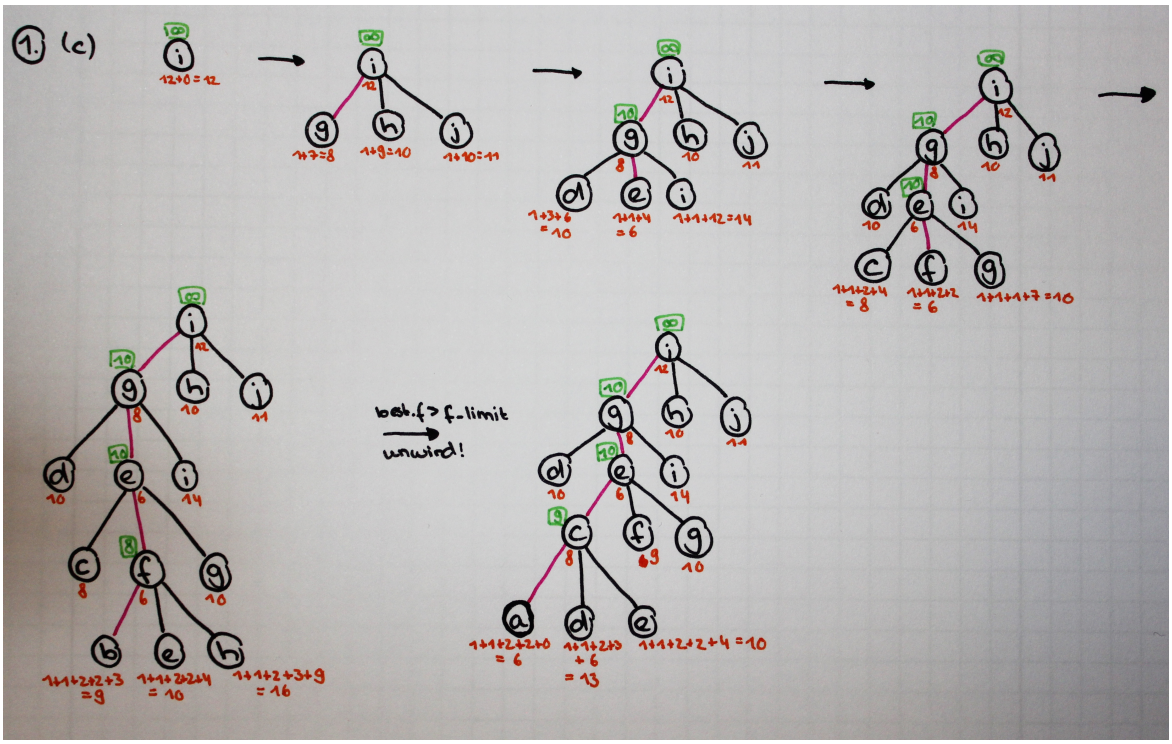
(a)



(b)



(c)



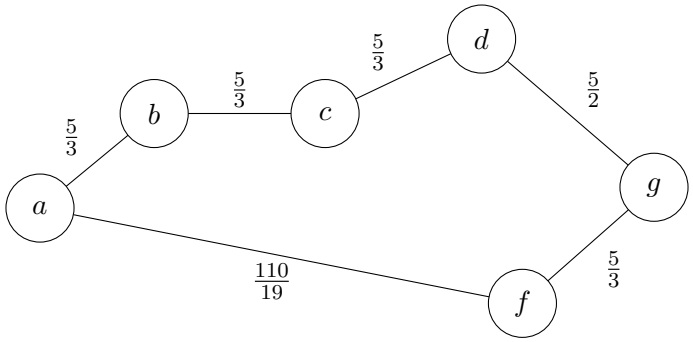
Question 2

The heuristic $h(n) = 0$ is monotone, therefore we are using graph search A* ($\rightarrow$ marking visited nodes)

Table of time $t = \frac{distance}{\varnothing velocity}$ in h

d \ transp.	car	train	fast train
200 km	$\frac{200}{110} \approx \frac{20}{11}$	$\frac{200}{120} \approx \frac{5}{3}$	/
300 km	$\frac{300}{110} \approx \frac{30}{11}$	$\frac{300}{120} \approx \frac{5}{2}$	/
200 km	$\frac{1100}{110} = 10$	/	$\frac{1100}{190} \approx \frac{110}{19}$

Graph with time, only going by train:



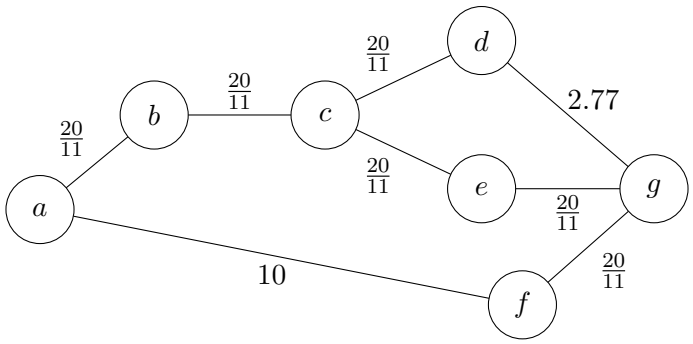
The table contains the minimal calculated *f(n)*-values at each step for node n. The purple colored *f(n)*-value is the one with minimal costs for the subpath and will be expanded in the next step. The gray values are values of already visited nodes, which do not occur anymore in a priority queue.

Table of f-Values: ($f(n) = \sum(g(n))$, since $h(n) = 0$)

	a	**b**	**c**	**d**	**f**	**g**
0	0	/	/	/	/	/
1	(0)	$\frac{5}{3} \approx 1,67$	/	/	$\frac{110}{19} \approx 5.79$	/
2	(0)	($\frac{5}{3} \approx 1,67$)	$\frac{10}{3} \approx 3.33$	/	$\frac{110}{19} \approx 5.79$	/
3	(0)	($\frac{5}{3} \approx 1,67$)	($\frac{10}{3} \approx 3.33$)	$\frac{15}{3} \approx 5$	$\frac{110}{19} \approx 5.79$	/
4	(0)	($\frac{5}{3} \approx 1,67$)	($\frac{10}{3} \approx 3.33$)	($\frac{15}{3} \approx 5$)	$\frac{110}{19} \approx 5.79$	$\frac{15}{2} \approx 7.5$
5	(0)	($\frac{5}{3} \approx 1,67$)	($\frac{10}{3} \approx 3.33$)	($\frac{15}{3} \approx 5$)	$\frac{110}{19} \approx 5.79$	$\frac{110}{19} + \frac{5}{3} \approx 7.46$

$\Rightarrow$ Travelling from city a to city g by train takes approximately 7.46 hours.

Graph with time, only going by car:



The table contains the minimal calculated *f(n)*-values at each step for node n. The purple colored *f(n)*-value is the one with minimal costs for the subpath and will be expanded in the next step. The gray values are values of already visited nodes, which do not occur anymore in a priority queue.

Table of f-Values: ($f(n) = \sum(g(n))$, since $h(n) = 0$)

	a	**b**	**c**	**d**	**e**	**f**	**g**
0	0	/	/	/	/	/	/
1	(0)	$\frac{20}{11} \approx 1.82$	/	/	/	10	/
2	(0)	($\frac{20}{11} \approx 1.82$)	$\frac{40}{11} \approx 3,63$	/	/	10	/
3	(0)	($\frac{20}{11} \approx 1.82$)	($\frac{40}{11} \approx 3,63$)	$\frac{60}{11} \approx 5.45$	$\frac{60}{11} \approx 5.45$	10	/
4	(0)	($\frac{20}{11} \approx 1.82$)	($\frac{40}{11} \approx 3,63$)	($\frac{60}{11} \approx 5.45$)	$\frac{60}{11} \approx 5.45$	10	$\frac{90}{11} \approx 8.18$
5	(0)	($\frac{20}{11} \approx 1.82$)	($\frac{40}{11} \approx 3,63$)	($\frac{60}{11} \approx 5.45$)	($\frac{60}{11} \approx 5.45$)	10	$\frac{80}{11} \approx 7.27$

⇒ Travelling from city a to city g by car takes approximately 7.27 hours.
⇒ Travelling by car (7.27 h) is faster than by train (7.46 h).

Question 3

(a)

(b)

RBFS return (348 (KARLSRUHE MANNHEIM KOBLENZ AACHEN) (AACHEN KOBLENZ MANNHEIM KARLSRUHE))) for the path from Karlruhe to Aachen. The list of visited nodes contains

(c)

The A*-search from Karlruhe to Aachen leads to *(348.0 (KARLSRUHE MANNHEIM KOBLENZ AACHEN) (KARLSRUHE MANNHEIM KOBLENZ TRIER FRANKFURT/MAIN))*. The list of visited nodes contains 5 cities.

The A*-search from Trier to Dresden leads to *(961.0 (TRIER KOBLENZ KOELN MUENSTER OSNABRUECK HANNOVER MAGDEBURG LEIPZIG DRESDEN) (TRIER MANNHEIM KOBLENZ SAARBRUECKEN FRANKFURT/MAIN FULDA KARLSRUHE WUERZBURG STUTTGART KOELN BAYREUTH NUERNBERG KASSEL ULM ERFURT ESSEN AUGSBURG CHEMNITZ AACHEN MUENSTER REGENSBURG OSNABRUECK MUENCHEN FREIBURG HANNOVER MAGDEBURG LEIPZIG))*. The list of visited nodes contains 27 cities.

The A*-search from Passau to Wilhelmshaven leads to the following output:
(1140.0 (PASSAU REGENSBURG NUERNBERG WUERZBURG FRANKFURT/MAIN KOBLENZ KOELN MUENSTER OSNABRUECK BREMEN WILHELMSHAVEN) (PASSAU REGENSBURG NUERNBERG BAYREUTH WUERZBURG FULDA KASSEL FRANKFURT/MAIN MUENCHEN AUGSBURG ULM KOBLENZ STUTTGART KOELN ESSEN GARMISCH-PART. MANNHEIM ERFURT MUENSTER KARLSRUHE OSNABRUECK AACHEN BREMEN)) . The list of visited nodes contains 23 cities.

Unfortunately we couldn't compile the code of the RBFS-algorithm. Though we assume, that RBFS visits more nodes than A*. RBFS expands a branch of the search tree as long as new nodes have an f-value smaller than the f_limit. Therefore the branch might be expanded for several recursive steps without knowing whether it's the right branch leadning to the goal. Unwinding and expansion of another tree leads to numerous more nodes that will be visited until the goal is reached. Therefore RBFS visits much more nodes than A*.