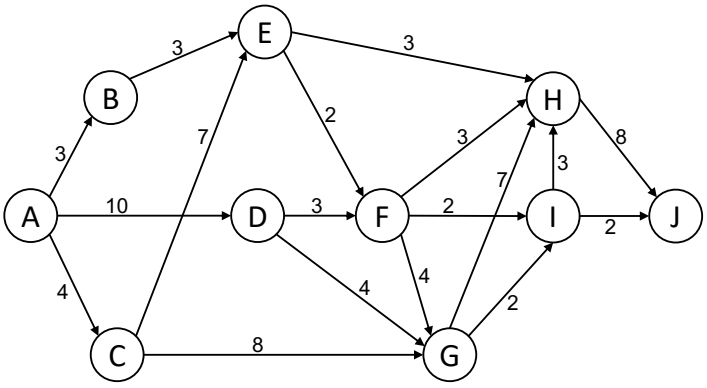
Solutions to Problem Sheet 7

1. Consider the network below with road lengths indicating the distance between each city. Use dynamic programming to find the shortest path from A to J.



Solution 1 *First order the cities by the maximum number of links between itself and J: JHIGFEDCBA. Then do value iteration:*

From	to	r	$r+f$	f
J	$\star$	0	0	0
H	J	8	8	8
I	J	2	2	
	H	3	11	2
G	H	7	15	
	I	2	4	4
F	H	1	9	
	I	2	4	
	G	4	6	4
E	H	3	11	
	F	2	6	6
D	F	3	7	
	G	4	8	7
C	E	10	16	
	G	8	12	12
B	E	3	9	9
A	B	3	12	
	C	4	16	
	D	10	17	12

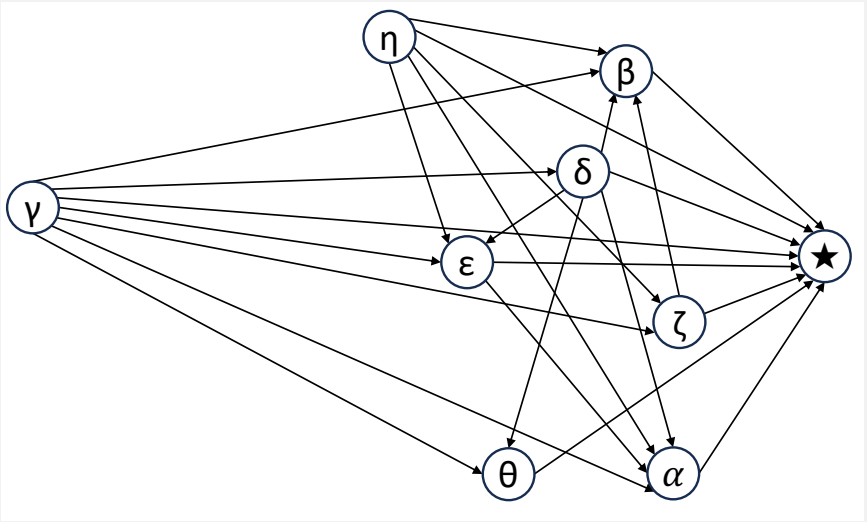
And so the length of the shortest path between A and J is 12. The optimal path is ABEFIJ.

2. You have eight cardboard boxes in your house with the following dimensions:

Box	Length	Width	Height
α	10	10	10
β	15	6	6
γ	3	3	3
δ	4	5	7
ϵ	9	9	11
ζ	11	4	4
η	5	3	4
θ	5	10	6

They are fragile and cannot be rotated of flipped. A box can be stacked on top of another box if its length is shorter and its width is shorter than the other box. Use dynamic programming to find the tallest possible stack of boxes.

Solution 2 *Draw the directed acyclic graph, with an edge between two boxes if one can be stacked on top of the other:*



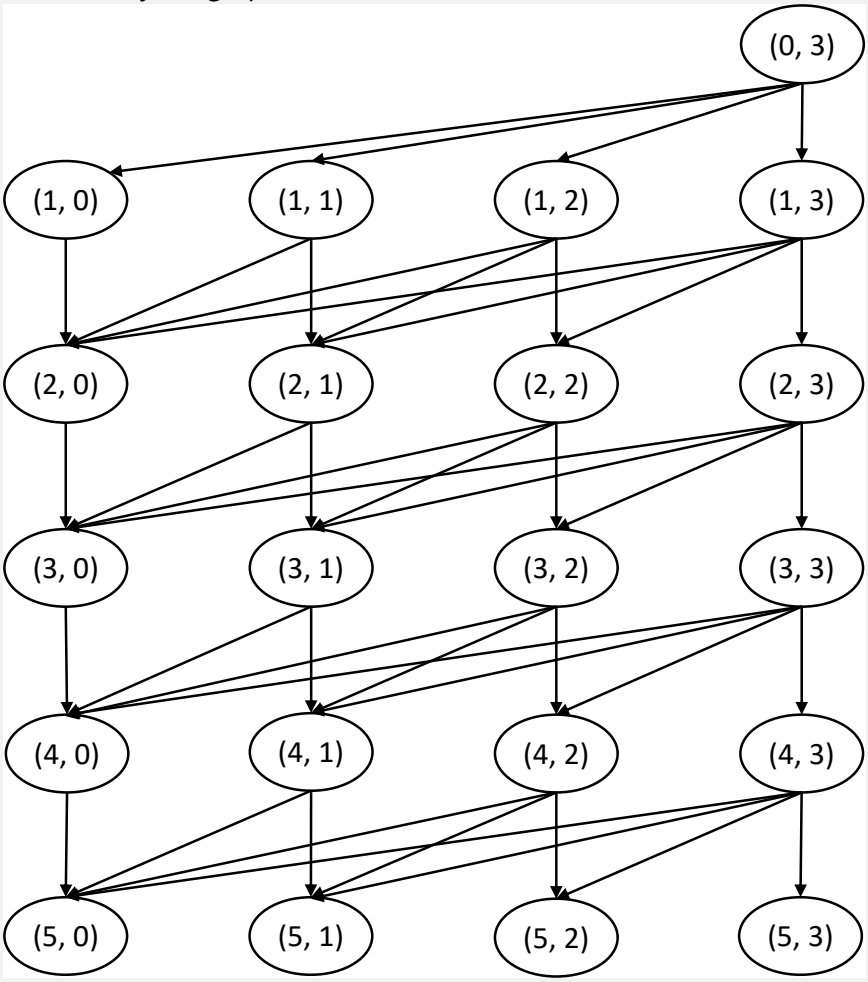
Solution 2 (continuing from p. 2) *Then do value iteration:*

Box	*Under*	r	$r+f$	f
$\star$	-	*0*	*0*	*0*
α	$\star$	*10*	*10*	*10*
β	$\star$	*6*	*6*	*6*
θ	$\star$	*6*	*6*	*6*
ϵ	$\star$	*11*	*11*	
	α	*11*	*21*	*21*
δ	$\star$	*7*	*7*	
	α	*7*	*17*	
	β	*7*	*13*	
	ϵ	*7*	*28*	
	θ	*7*	*13*	*28*
ζ	$\star$	*4*	*4*	
	β	*4*	*10*	*10*
γ	$\star$	*3*	*3*	
	α	*3*	*10*	
	β	*3*	*9*	
	δ	*3*	*31*	
	ϵ	*3*	*24*	
	ζ	*3*	*13*	
	θ	*3*	*9*	*31*
η	$\star$	*4*	*4*	
	α	*4*	*14*	
	β	*4*	*10*	
	ϵ	*4*	*25*	
	ζ	*4*	*14*	*25*

And so the length of the tallest stack is 31. The optimal stack is $\alpha - \epsilon - \delta - \gamma$.

3. A mobile burger truck has been contracted to supply burgers for a music festival. The festival is to last 5 days, however the burger truck only has 3 boxes of frozen meat, and once defrosted each box of frozen meat only lasts one day. It costs €$1k$ per box per day to keep frozen. The burger truck must decide how many boxes to defrost and sell each day of the festival. They anticipate that they will take in €$1.5k$ per day if they defrost one box, €$3k$ if they defrost two boxes on the same day, and €$3.5k$ if they defrost 3 boxes on the same day. However, due to a broken contract, they will incur a fine of €$2k$ each day they do not offer burgers. Use dynamic programming to devise a strategy that will maximise the burger truck's intake.

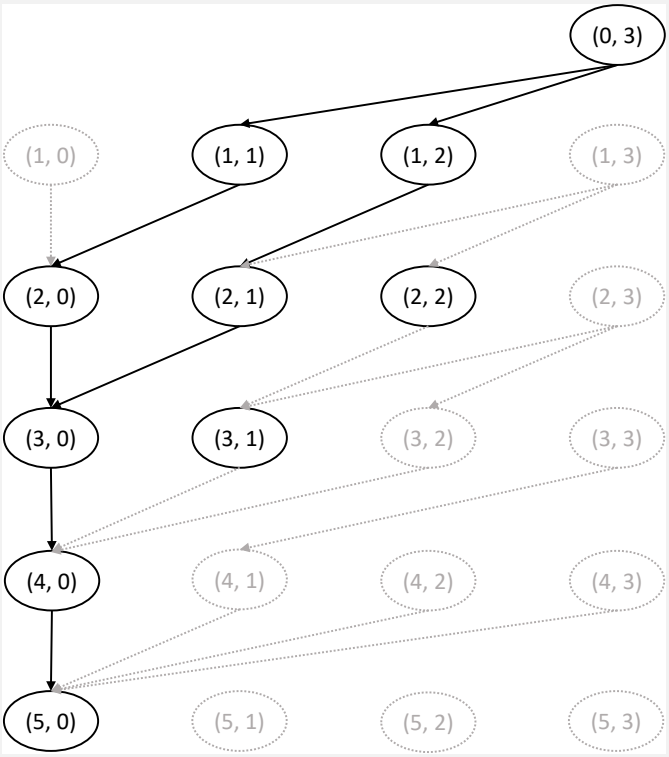
Solution 3 *Let (t, n) be a state denoting that there are n frozen boxes left at then end of day t. We get the following directed acyclic graph:*



Solution 3 (continuing from p. 4) *Performing value iteration:*

(t, n)	To defrost	r	$r + f$	f
(5, 3)	-	0	0	0
(5, 2)	-	0	0	0
(5, 1)	-	0	0	0
(5, 0)	-	0	0	0
(4, 0)	0	-20	$-20 + 0 = -20$	-20
(4, 1)	0	-30	$-30 + 0 = -30$	
	1	15	$15 + 0 = 15$	15
(4, 2)	0	-40	$-40 + 0 = -40$	
	1	5	$5 + 0 = 5$	
	2	30	$30 + 0 = 30$	30
(4, 3)	0	-50	$-50 + 0 = -30$	
	1	-5	$-5 + 0 = -5$	
	2	20	$20 + 0 = 20$	
	3	35	$35 + 0 = 35$	35
(3, 0)	0	-20	$-20 - 20 = -20$	-40
(3, 1)	0	-30	$-30 + 15 = -15$	
	1	15	$15 - 20 = -5$	-5
(3, 2)	0	-40	$-40 + 35 = -5$	
	1	5	$5 + 15 = 20$	
	2	30	$30 - 20 = 10$	20
(3, 3)	0	-50	$-50 + 35 = -15$	
	1	-5	$-5 + 30 = 25$	
	2	20	$20 + 15 = 35$	
	3	35	$35 - 20 = 15$	35
(2, 0)	0	-20	$-20 - 40 = -60$	-60
(2, 1)	0	-30	$-30 - 50 = -35$	
	1	15	$15 - 40 = -25$	-25
(2, 2)	0	-40	$-40 + 20 = -20$	
	1	5	$5 - 5 = 0$	
	2	30	$30 - 40 = -10$	0
(2, 3)	0	-50	$-50 + 35 = -15$	
	1	-5	$-5 + 20 = 15$	
	2	20	$20 - 5 = 15$	
	3	35	$35 - 40 = -5$	15
(1, 0)	0	-20	$-20 - 60 = -80$	-80
(1, 1)	0	-30	$-30 - 25 = -55$	
	1	15	$15 - 60 = -45$	-45
(1, 2)	0	-40	$-40 + 0 = -40$	
	1	5	$5 - 25 = -20$	
	2	30	$30 - 60 = -30$	-20
(1, 3)	0	-50	$-50 + 15 = -35$	
	1	-5	$-5 + 0 = -5$	
	2	20	$20 - 25 = -5$	
	3	35	$35 - 60 = -25$	-5
(0, 3)	0	-50	$-50 + 5 = -45$	
	1	-5	$-5 - 20 = -25$	
	2	20	$20 - 45 = -25$	
	3	35	$35 - 80 = -50$	-25

Solution 3 (continuing from p. 5) *To help read off the solution, redraw the DAC but only keeping the optimal actions, and shading out all optimal actions that are unreachable from the beginning state (0, 3).*



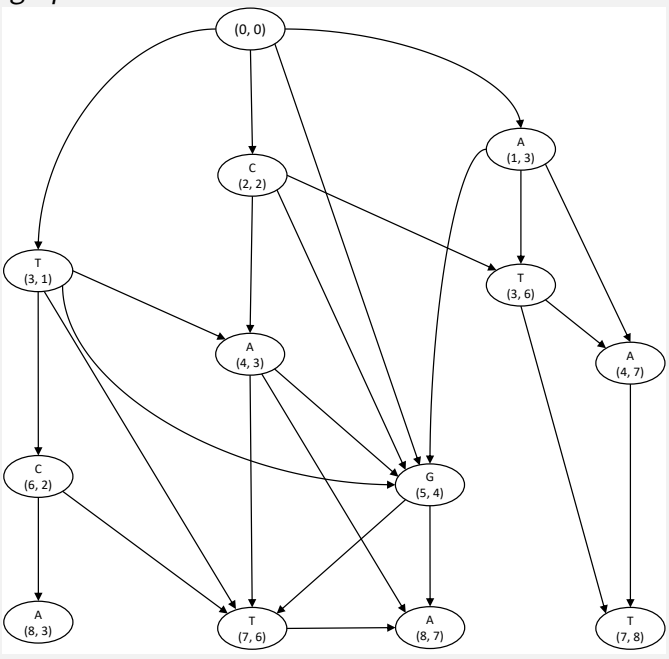
Therefore the optimal strategy is to either:

- *defrost two boxes on the first day, one on the second day, and zero thereafter, or*
- *defrost one box on the first day, one on the second, one on the third, and zero thereafter.*

4. Finding the longest common subsequence between to sequences is a useful problem to solve, for example it can tell us how similar two strands of DNA are. Use dynamic programming to find the longest common subsequence between ACTAGCTA and TCAGGTAT.

   (Hint: consider the triples (i, j, k) as states, corresponding to the i^{th} element of the first sequence, and the j^{th} element of the second sequence. You only need to consider states where these letters, k, are identical. E.g. the state (1, 3, A) denoted the letter A is shared by the first letter of the first sequence, and the third letter of the second sequence.)

Solution 4 *Let (i, j, k) denote a state corresponding to the i^{th} element of the first sequence, and the j^{th} element of the second sequence. Let us only consider states where these elements are idenitical, k. E.g. (1, 3, A) corresponds to the letter A being the 1st element of the first sequence and the 3rd letter of the 2nd sequence.*
All states are given by: (1, 3, A), (1, 7, A), (4, 3, A), (4, 7, A), (8, 3, A), (8, 7, A), (2, 2, C), (6, 2, C), (3, 1, T), (3, 6, T), (3, 8, T), (7, 1, T), (7, 6, T), (7, 8, T), (5, 4, G), (5, 5, G).
We can draw directed acyclic graph representing whether each letter can preceed the other in the subsequence. Order the states, that is $s_1 = (i_1, j_1, k_1)$ preceeds $s_2 = (i_2, j_2, k_2)$ if $i_1 < i_2$ and $j_i < j_2$. Draw an edge from s_1 to s_2 if s_1 preceeds s_2.
Note that we can simplify the problem by noting that there is no point skipping valid letters from the subsequence. So, from each state s, for each element k only draw one edge from s to another state with element k, and ensure that this state is the first such state in the ordering. This means some states become unused.
Draw the directed acyclic graph:



Solution 4 (continuing from p. 7) *Now do value iteration, noting that each edge adds one element to the subsequence:*

From	to	$r+f$	f
(0, 0, 0)	$\star$	0	0
(1, 3, A)	(0, 0, 0)	1	1
(2, 2, C)	(0, 0, 0)	1	1
(3, 1, T)	(0, 0, 0)	1	1
(3, 6, T)	(1, 3, A)	2	
	(2, 2, C)	2	2
(4, 3, A)	(2, 2, C)	2	
	(3, 1, T)	2	2
(4, 7, A)	(1, 3, A)	2	
	(3, 6, T)	3	3
(5, 4, G)	(0, 0, 0)	1	
	(1, 3, A)	2	
	(2, 2, C)	2	
	(3, 1, T)	2	
	(4, 3, A)	3	3
(6, 2, C)	(3, 1, T)	2	2
(7, 6, T)	(3, 1, T)	2	
	(4, 3, A)	3	
	(5, 4, G)	4	
	(6, 2, C)	3	4
(7, 8, T)	(3, 6, T)	3	
	(4, 7, A)	4	4
(8, 3, A)	(6, 2, C)	3	3
(8, 7, A)	(4, 3, A)	3	
	(5, 4, G)	4	
	(7, 6, T)	5	5

And so the length of the longest common subsequence is 5, obtained by: $(0,0,0) \to (3,1,T) \to (4,3,A) \to (5,4,G) \to (7,6,T) \to (8,7,A)$ *or by* $(0,0,0) \to (2,2,C) \to (4,3,A) \to (5,4,G) \to (7,6,T) \to (8,7,A)$. *That is: TAGTA or CAGTA.*