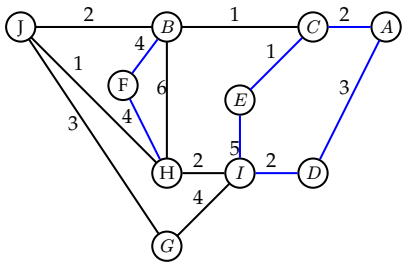
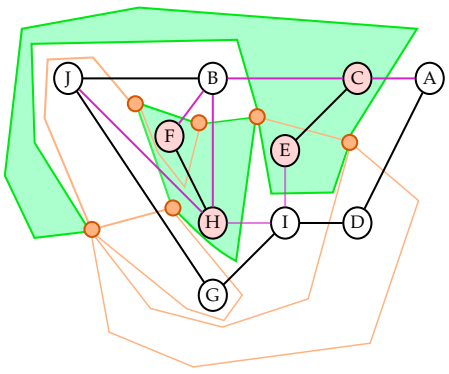
🔓 Answer keys for the final exam 🗓️ 2024 - 2025

Important remark : if an answer B depends on an answer A, then the mark B can only be given if answer A is correct.

📁 Exercise 1

No.	Answers	Points
	Part 1	
1.	A piece of information will be discarded after it has been sent and received back by a given user. This corresponds to a cycle. To reach everyone in the network before being discarded, the cycle needs to be Hamiltonian.	0.5
	To check whether the graph is Hamiltonian, we use the following property: the edges that are incident to vertices with degree 2 in a Hamiltonian graph belong to the Hamiltonian cycle. Using this property, the edge in blue should belong to the Hamiltonian cycle (if it exists). However, we can see that this will result in a cycle (A,C,E,I,D,A), therefore the graph is not Hamiltonian (a piece of information cannot be discarded after visiting everyone the in network once). 	1
2.	This graph is not bipartite.	0.5
	The graph contains a cycle with a length of 5 (A,C,E,I,D,A). A bipartite graph does not have a cycle with an odd length.	1
3.	One possible solution is to use the co-cycle of $\{E, C, F, H\}$. This can be done by looking for the smallest cycle in the dual graph that contains these vertices.	0.5
	The green region is the smallest cycle in the dual graph, which includes $\{E, C, F, H\}$. The edges to be removed are shown in purple. Their number is 7 (which corresponds the length of the cycle in the dual graph). 	1

4. The problem can be solved by graph coloring, since two adjacent vertices (two persons that know each other) will have different colors (the persons won't belong to the same team). We compute the chromatic number δ. **0.5**

Upper bounds:
1. $\delta \leq 4$ (the graph is planar)
2. $\delta \leq 4$ (the graph's degree)
3. $\delta \leq 10 + 1 - 4 = 7$ (the maximum stable is $\{A, E, F, J\}$)

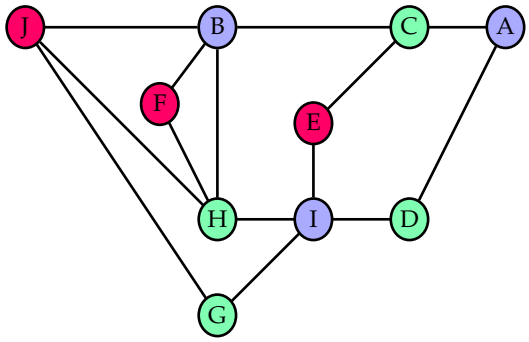
Lower bounds:
1. $\delta \geq 3$ (the graph is not bipartite)
2. $\delta \geq 3$ (the maximum clique has 3 vertices)

We have: $3 \leq \delta \leq 4$.

Let's apply the Welsh and Powell's algorithm: **1.5**

Vertex	B	H	I	C	J	A	D	E	F	G
Degree	4	4	4	3	3	2	2	2	2	2
Init	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
it 1	Blue	N/A	Blue	N/A	N/A	Blue	N/A	N/A	N/A	N/A
it 2	Blue	Green	Blue	Green	N/A	Blue	Green	N/A	N/A	Green
it 3	Blue	Green	Blue	Green	Red	Blue	Green	Red	Red	Green

We get the following graph (it is not required to draw this graph):



The minimum number of teams is 3. **0.5**

5. Many solutions may be used here. One of them is to build a graph in which two users i and j are adjacent if there is a walk from i to j with length ≤ 2 (with $i \neq j$). We get the following graph (defined by its lists of adjacency):

Vertex	**Adjacent vertices**	**Vertex**	**Adjacent vertices**
A	B,C,D,E,I	B	A,C,E,F,G,H,I,J
C	A,B,D,E,F,H,I,J	D	A,C,E,G,H,I
E	A,B,C,D,G,H,I	F	B,C,H,I,J
G	B,D,E,H,I,J	H	B,C,D,E,F,G,I,J
I	A,B,C,D,E,F,G,H,J	J	B,C,F,G,H,I

Now, by using the Welsh and Powell's algorithm (not required), we get: 0.5

Vertex	I	B	C	H	E	D	G	J	A	F
Degree	9	8	8	8	7	6	6	6	5	5
Init	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
it 1	Blue	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
it 2	Blue	Green	N/A	N/A	N/A	Green	N/A	N/A	N/A	N/A
it 3	Blue	Green	Red	N/A	N/A	Green	Red	N/A	N/A	N/A
it 4	Blue	Green	Red	Yellow	N/A	Green	Red	N/A	Yellow	N/A
it 5	Blue	Green	Red	Yellow	Purple	Green	Red	Purple	Yellow	N/A
it 6	Blue	Green	Red	Yellow	Purple	Green	Red	Purple	Yellow	Gray

The minimum number of teams is 6. 0.5

Part 2

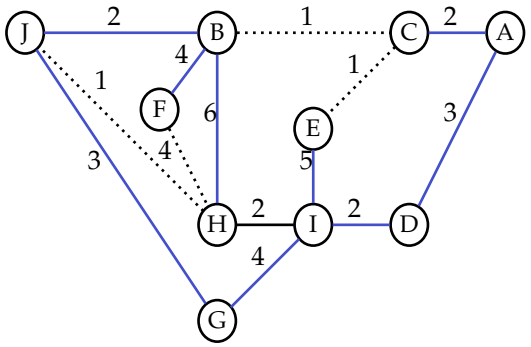
1. The new network is a spanning tree. 0.5

Justification: since every user receives the information once, there is one walk between any two users. 0.5

2. We use here the Kruskal's algorithm to build the maximum spanning tree. 2.5

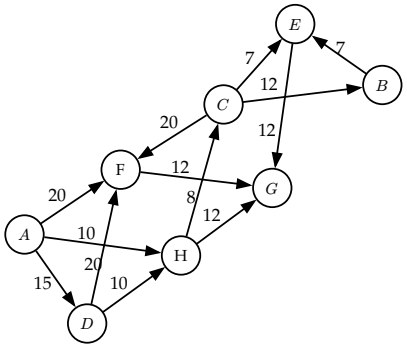
Edge	(B,H)	(E,I)	(B,F)	(I,G)	(F,H)	(A,D)	(J,G)	(A,C)	(B,J)	(D,I)	(H,I)	(B,C)	(C,E)	(H,J)
Weight	6	5	4	4	4	3	3	2	2	2	2	1	1	1
Belongs	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗

The maximum spanning tree is 0.5



Exercise 2

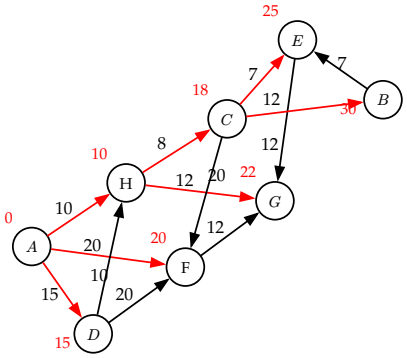
No.	Answers	Points
1.	Each page is modelled by a vertex and every hyperlink is modelled as an arc. If the loading time of a page X is t, then all ingoing arcs to X are labelled with t. We get the following graph:	1



2. This question is about using building the arborescence of the shortest paths in this graph. The problem can be handled with the Dijkstra's algorithm or the Bellman-Ford's algorithm (both are accepted here). **3**

S	λA	λB	λC	λD	λE	λF	λG	λH
A	0	∞	∞	15	∞	20	∞	10
A,H	0	∞	18	15	∞	20	22	10
A,H,D	0	∞	18	15	∞	20	22	10
A,H,D,C	0	30	18	15	25	20	22	10
A,H,D,C,F	0	30	18	15	25	20	22	10
A,H,D,C,F,G	0	30	18	15	25	20	22	10
A,H,D,C,F,G,E	0	30	18	15	25	20	22	10

The arborescence of the shortest paths is: **1**



3.a The constraint means that if there is a path from a page X to a page Y and a path from Y to a page Z, there should also be a path from X to Z. This corresponds to a transitive graph. If the graph is not transitive, we can achieve transitivity by building the transitive closure. **1.5**

3.b It is not required to apply the algorithm presented in the course. We will simply add the required arcs manually using the following rule: if there is an arc from X to Y and an arc from Y to Z, then add an arc from X to Z if it does not already exist (we removed the weights since they are irrelevant for this question). The minimum number of hyperlinks to add is 13. **1.5**