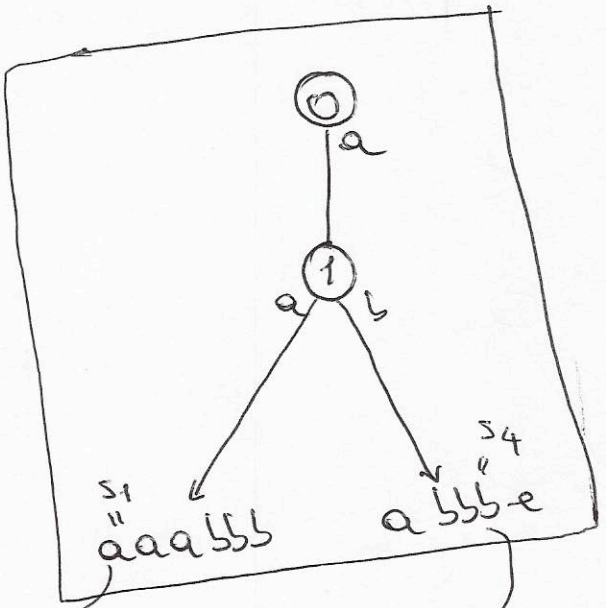
Solution FinalTerm : 17 December 2019

Exercise 1

FC = <0,baci><3,0><1,oss><1,vono><2,u><0,conti><4,ola>

LPFC = <0,baci><0,baco><0,boss><1,vono><0,suv>
<0,conti><4,ola>

Exercise 2



Two-level indexing is described in the notes: 9-5, 9-12

<0, ~~aaabbb~~><4,bd><1,bbbd>

<0, ~~abbbe~~><3,c><2,c

The search for the string P = aaabcb consists of 3 phases:

1- downward: It matches the first two "a" of P and reaches the first leaf.

2- check: It computes LCP(P, aaabbb) = 4 & P > s_1

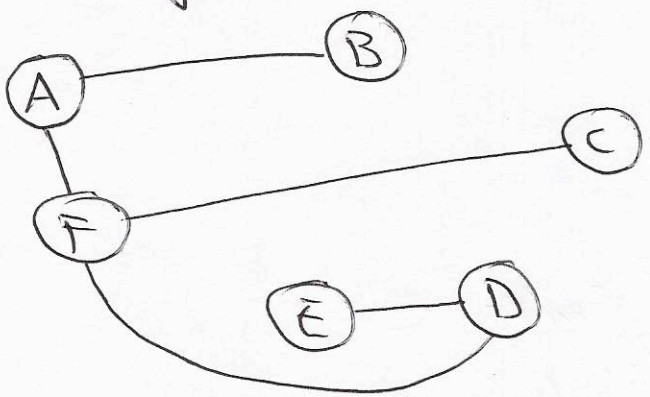
3- upward: It finds the position of the mismatch, which is on the leftmost downward edge and since P > s_1 it declares the lexicographic position of P between s_1 and s_4.

Thus it accesses the first block on disk, and finds that P occurs between s_1 and s_2, by decompressing the front-coded strings of that block in 1 I/O.

Exercise 3

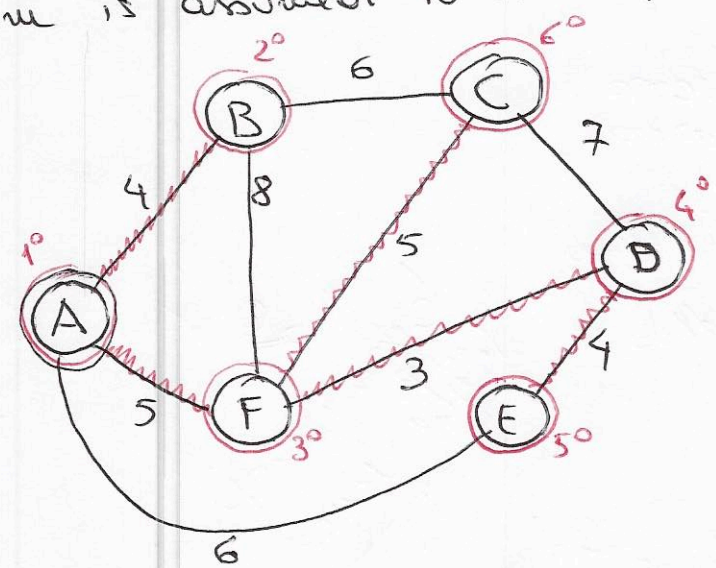
- Kruskal sorts edges by their increasing weight

(D, F, 3)	V
(A, B, 4)	V
(D, E, 4)	V
(A, F, 5)	V
(C, F, 5)	V
(A, E, 6)	X
(B, C, 6)	X
(C, D, 7)	X
(B, F, 8)	X



MST built by inserting edges in increasing weight and guaranty that no cycle is formed, hence no edge internal to a component.

- Prim is assumed to start from node A



	A	B	C	D	E	F
	0	∞	∞	∞	∞	∞
1°	V	4_A	∞	∞	6_A	5_A
2°		V	6_B	∞	6_A	5_A
3°			5_F	3_F	6_A	V
4°			5_F	V	4_D	
5°			V		V	

The MST is the same one computed by Kruskal's algorithm.