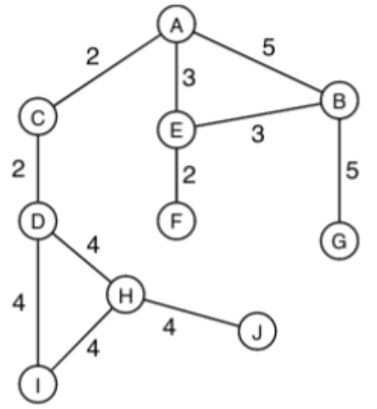
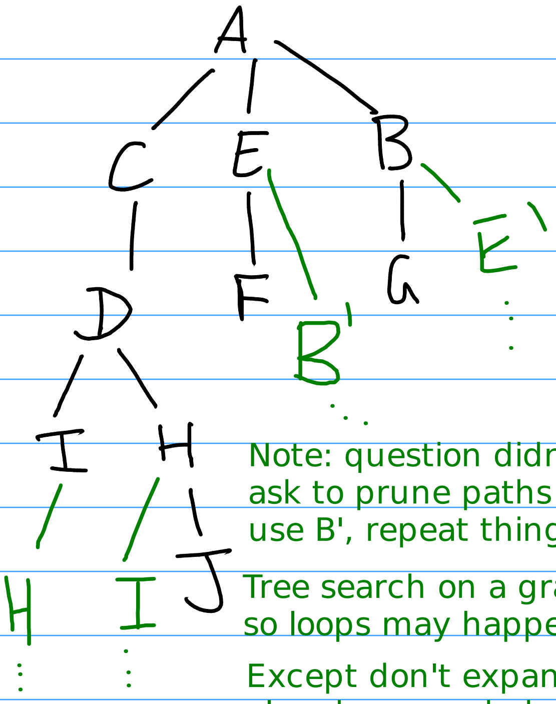
1)



i)



Note: question didn't ask to prune paths - use B', repeat things etc

Tree search on a graph so loops may happen!

Except don't expand already expanded ones? Not too sure

ii)

1. A(0)
2. C(2), E(3), B(5)
3. E(3), D(4), B(5)
4. D(4), B(5), F(5), ~~B(6)~~
5. B(5), F(5), H(8), I(8)
6. F(5), H(8), I(8), G(10)
7. H(8), I(8), G(10)
8. I(8), G(10), ~~I(12)~~, J(12)
9. G(10), ~~H(12)~~, J(12)
10. J(12) Solution found

iii) ACEDBFHIGJ 9 nodes expanded.

iv) Route: ACDHJ Cost: 12

I think what I did is correct for Q2, graph search instead of tree search, though graph search includes a closed list + frontier is paths, not nodes

3) a)

	Frontier	Closed
1.	A(10)	
2.	E(5), B(6), C(8)	A
3.	F(3), B(6), C(8), <Pruned AEB>	AE
4.	B(6), C(8)	AEF
5.	G(1), C(8)	AEFB
6.	C(8)	AEFBG
7.	D(5)	AEFBGC
8.	H(3), I(4)	AEFBGCD
9.	J(0), I(4)	AEFBGCDH
10.	Solution found @ J.	

Note: should be pruned later, once AEB is next to be expanded.
Check if nodes in path are in closed list - if so, prune the path.

Route A C D H J cost 12.

	b) Frontier	Closed
1.	A(10)	
2.	E(8), C(10), B(11)	A
3.	F(8), C(10), B(11), <pruned AEB(12)>	AE
4.	C(10), B(11)	AEF
5.	D(9), B(11)	AEFC
6.	B(11), H(11), I(12)	AEFCD
7.	H(11), G(11), I(12)	AEFCDB
8.	G(11), I(12), J(12)	AEFCDBH
9.	I(12), J(12)	AEFCDBHG
10.	J(12), <pruned ACDIH(15)>	AEFCDBHGI

11. Solution found @J.
Path: ACDHJ cost: 12

4) It is not — h_2 is not admissable since
$\text{MinCost}(D \rightarrow J) = 8 < h(D) = 10$.

5) H_2 is best, since it is guaranteed to dominate all $h_1 \dots h_n$.
H_1, H_2 ~~& H_3~~ are admissable.

H3 is not guaranteed to be admissible, it could be an overestimate since it is a sum.