1 Question 1

Show that the variable elimination on polytrees has computational complexity that is linear in the size of the network; size is defined in terms of the number of entries in the CPTs so the cost grows exponentially with the number of parents of a node. As for any other graph structure, the complexity depends on elimination order. The task is to derive an algorithm that achieves linear complexity (as defined above) and applies to any given polytree.

Solution:
Let $\mathcal{B}$ be the Bayesian network in question (which is a polytree). Let $\mathcal{T}$ be the undirected skeleton of $\mathcal{B}$. Since $\mathcal{B}$ is a polytree, $\mathcal{T}$ must necessarily be a tree. The algorithm is as follows: at each step, we choose any leaf l in undirected tree $\mathcal{T}$ (a node whose degree is 1) to eliminate. Note that the resulting network (after eliminating leaf l) is also a polytree. Thus, there will always be an available leaf, and the algorithm will never become stuck. Using this algorithm, we can guarantee that all intermediate factors generated have scopes involving only subsets of the scopes of the factors in the original network. To understand why, consider the two possible cases for leaf l :

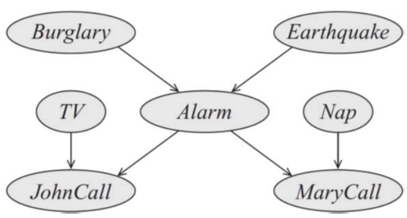
- The eliminated leaf l is a parent of one child c (and has no parents of its own). When eliminating l we eliminate two factors - $\phi(l)$ and $\phi(c, \mathbf{Pa}_c)$ and create an intermediate factor $\psi(c, \mathbf{Pa}_c)$. Note that $\mathbf{Pa}_c$ represents the parents of c in $\mathcal{B}$ and $l \in \mathbf{Pa}_c$.

- The eliminated leaf l is a child with no children and one parent p. We eliminate factor $\phi(l, p)$ and generate intermediate factor $\psi(l, p)$.

VE is linear in the size of the largest intermediate factor and the number of variables, $O(nN^*_{max})$, where n is the number of original CPT's (one for each of n variables) and N^*_{max} is the number of entries in the largest intermediate factor. Above, we showed that the largest intermediate factor is no larger than any original factor (CPT) of the network. The result is proved.

2 Question 2

Consider the "Alarm" network of Fig 3.14 (reproduced below with CPT's). Now compute probability of Earthquake given that JohnCalls is True and MaryCalls is False. Use the Variable Elimination algorithm described in the book (section 9.3, Algorithm 9.2). Solution:



P (B = T) = 0.002; P (E = T) = 0.001; P (TV = T) = 0.8; P (NAP = T) = 0.25.

B	E	P(A) = T
T	T	0.93
T	F	0.85
F	T	0.6
F	F	.002

TV	Alarm	P(JohnCalls) = T
T	T	0.45
T	F	0.03
F	T	0.88
F	F	.02

NAP	Alarm	P(MaryCalls) = T
T	T	0.25
T	F	0.001
F	T	0.75
F	F	.01

Factors involved (reduced to evidence):
$\phi_B(B)$, $\phi_A(A, B, E)$, $\phi_E(E)$, $\phi_T(T)$, $\phi_N(N)$,$\phi_J[j^1](T, A)$, $\phi_M[m^0](N, A)$

- Step 1, eliminate B:
  $\psi_1(A, B, E) = \phi_B(B)\phi_A(A, B, E)$
  $\tau_1(A, E) = \Sigma_B \psi_1(A, B, E)$
- Step 2, eliminate T:
  $\psi_2(T, A) = \phi_T(T)\phi_J[j^1](T, A)$
  $\tau_2(A) = \Sigma_T \psi_2(T, A)$
- Step 3, eliminate N:
  $\psi_3(N, A) = \phi_N(N)\phi_M[m^0](N, A)$
  $\tau_3(A) = \Sigma_N \psi_3(T, A)$
- Step 4, eliminate A:
  $\psi_4(A, E) = \tau_1(A, E)\tau_2(A)\tau_3(A)$
  $\tau_4(E) = \Sigma_A \psi_4(A, E)$
- Step 5, multiply factors containing E and normalize:
  $\psi_5(E) = \phi_E(E)\tau_4(E)$

The answer then is:
$P(E = e^1|j^1, m^0) = 0.00462$
$P(E = e^0|j^1, m^0) = 0.99538$