6.438 Fall 2020 Problem Set 3

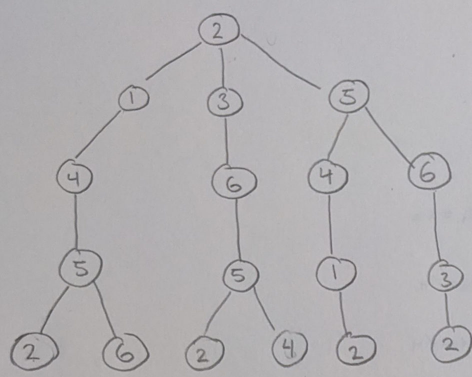
Rylan Schaeffer

Problem 1

1ab

3.1

a)



b) Recall that $m_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\, \psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus\{j\}} m_{k\to i}(x_i)$

We initialize all messages at boundary as 1 except the two givens: $\psi_2(x_2) = \begin{cases} 1-\gamma & x_2 = 0 \\ \gamma & x_2 = 1 \end{cases}$ $\quad \psi_5(x_5) = \begin{cases} \gamma & x_5 = 0 \\ 1-\gamma & x_5 = 1 \end{cases}$

Note that since edge potential for (i,j) is $\mathbb{1}_{x_i = x_j}$, we have transition operator $\begin{bmatrix} x_i = 0 \\ x_i = 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_j = 0 \\ x_j = 1 \end{bmatrix}$

0th Iteration) $P_2(x_2) = \psi_2(x_2) = \begin{bmatrix} 1-\gamma \\ \gamma \end{bmatrix}$

1st Iteration) $P(x_2) = \psi_2(x_2) \cdot \psi_5(x_5) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$

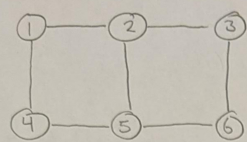
2nd Iteration) $P(x_2) = \psi_2(x_2) \cdot \psi_5(x_5) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$

3rd Iteration) $P(x_2) = \psi_2(x_2) \cdot \psi_5(x_5)\, \psi_5(x_5)\, \psi_5(x_5) \propto \begin{bmatrix} \gamma^2 \\ (1-\gamma)^2 \end{bmatrix}$

4th Iteration) $P(x_2) = \psi_2(x_2)\, \psi_5(x_5) \left(\psi_5(x_5)\right)^2 \left(\psi_2(x_2)\right)^4 \propto \begin{bmatrix} (1-\gamma)^2 \\ \gamma^2 \end{bmatrix}$

1c

3.1

c) 

Message update given by

$$m^{t+1}_{i\to j}(x_j) = \sum_{x_i} \phi_i(x_i)\,\psi_{ij}(x_i, x_j) \prod_{k\in N(i)\setminus\{j\}} m^{t}_{k\to i}(x_i)$$

Given: $m_{2\to 1} = \begin{bmatrix} \beta \\ 1-\beta \end{bmatrix}$ and $m_{2\to 3} = \begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix}$

Then

$$m_{1\to 4}(x_4) = \sum_{x_1} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} m_{2\to 1}(x_1)$$

$$= \begin{bmatrix} \beta \\ 1-\beta \end{bmatrix}$$

$$m_{4\to 5}(x_5) = \begin{bmatrix} \beta \\ 1-\beta \end{bmatrix}$$

Similarly

$$m_{3\to 6}(x_6) = \begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix}$$

$$m_{6\to 5}(x_5) = \begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix}$$

$$m_{5\to 2}(x_2) = \sum_{x_5} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \begin{bmatrix} (1-\gamma)\gamma & 0 \\ 0 & \gamma(1-\gamma) \end{bmatrix} \left(\begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix} \circ \begin{bmatrix} \beta \\ 1-\beta \end{bmatrix} \right)$$

$$\Rightarrow \boxed{m_{5\to 2}(x_2) = \begin{bmatrix} \alpha\beta \\ (1-\alpha)(1-\beta) \end{bmatrix}}$$

We backsolve for $m_{2\to 5}(x_5)$ from $p(x_5) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \begin{bmatrix} \alpha \\ 1-\alpha \end{bmatrix} \begin{bmatrix} \beta \\ 1-\beta \end{bmatrix} m_{2\to 5}(x_5)$

$$\Rightarrow m_{2\to 5}(x_5) = \begin{bmatrix} 1/(\alpha\beta) \\ 1/(1-\alpha)(1-\beta) \end{bmatrix}$$

Then backsolve for $m_{5\to 4}$ from $p(x_4) = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \begin{bmatrix} \beta \\ 1-\beta \end{bmatrix} m_{5\to 4}$

$$\Rightarrow m_{5\to 4}(x_4) = m_{4\to 1}(x_1) = m_{1\to 2}(x_2) = \begin{bmatrix} 1/\beta \\ 1/(1-\beta) \end{bmatrix}$$

Similarly $m_{5\to 6}(x_6) = m_{6\to 3}(x_3) = m_{3\to 2}(x_2) = \begin{bmatrix} 1/\alpha \\ 1/(1-\alpha) \end{bmatrix}$

$$\Rightarrow \boxed{\alpha, \beta \in (0, 1)}$$

(if α or β <0 or >1, then $p(x)$ is invalid. if α or $\beta = 1$ or 0, we have $1/\alpha$, $1/\beta$, $1/($ $1/(1-\beta)$ or $1/(1-\alpha)$ become infinite)

Problem 3

3ai

3.3)

a) i. $p(x,y) = p(y|x)p(x)$

$$= \exp\left(-\frac{1}{2}(y-Cx)^T R^{-1}(y-Cx)\right) \exp\left(-\frac{1}{2} x^T J_x x + h_x^T x\right)$$

$$= \exp\left(-\frac{1}{2} y^T R^{-1} y + x^T C^T R^{-1} y - \frac{1}{2} x^T C^T R^{-1} C x\right) \exp\left(-\frac{1}{2} x^T J_x x + h_x^T x\right)$$

$$= \exp\left(-\frac{1}{2} \begin{bmatrix} x \\ y \end{bmatrix}^T \underbrace{\begin{bmatrix} J_x + C^T R^{-1} C, & -C^T R^{-1} \\ -(R^{-1})^T C & R^{-1} \end{bmatrix}}_{J_{xy}} \begin{bmatrix} x \\ y \end{bmatrix} + \underbrace{\begin{bmatrix} h_x \\ 0 \end{bmatrix}}_{h_{xy}}^T \begin{bmatrix} x \\ y \end{bmatrix}\right)$$

a) ii. Here, I use properties of conditioning in precision form

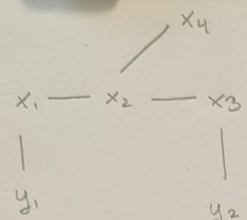
$$J_{x|y} = \left[J_{xy}\right]_{xx} = J_x + C^T R^{-1} C$$

$$J_{x|y} = h_x - J_{xy} y = h_x + C^T R^{-1} y$$

3aii

3.3

a) ii.



Given: $y_1 = x_1 + v_1$

$y_2 = x_3 + v_2$

$y = Cx + v \qquad v \sim N(0, R)$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

From 3.3 aii, we know that in general

$$\begin{bmatrix} x \\ y \end{bmatrix} \sim N^{-1}\left(\begin{bmatrix} h_x \\ 0 \end{bmatrix}, \begin{bmatrix} J_x + C^T R^{-1} C & -C^T R^{-1} \\ -(R^{-1})^T C & R^{-1} \end{bmatrix} \right)$$

and $x|y \sim N^{-1}\left(h_x + C^T R^{-1} y, \; J_x + C^T R^{-1} C \right)$

Here, $$x|y \sim N^{-1}\left(\underbrace{h_x + \begin{bmatrix} y_1 \\ 0 \\ y_2 \\ 0 \end{bmatrix}}_{h_{x|y}}, \; \underbrace{J_x + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{J_{x|y}} \right)$$

Thus

$$h_{x_3 \to x_2} = -J_{x_2 x_3} (J_{x_3 x_3})^{-1} h_{x_3}$$

$$= -\Big(\underbrace{J_x[2,3]}_{\text{2nd row, 3rd col of } J_x} \Big) \left(J_x[3,3] + 1 \right)^{-1} \Big(\underbrace{h_x[3]}_{\text{3rd element of } h_x} + y_2 \Big)$$

and $$J_{x_3 \to x_2} = -\left(J_x[2,3] \right) \left(J_x[3,3] + 1 \right)^{-1} \left(J_x[3,2] \right)$$

3aiii

3.3

a. iii)

$$\left.\begin{aligned} y_1 &= x_1 + v_1 \\ y_2 &= x_3 + v_2 \\ y_3 &= x_3 + v_3 \end{aligned}\right\} \Rightarrow \boxed{C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}$$

Generally $\quad h_{x|y} = h_x + C^T R^{-1} y$

$$J_{x|y} = J_x + C^T R^{-1} C$$

In this case, $\quad h_{x|y} = h_x + \begin{bmatrix} y_1 \\ 0 \\ y_2 + y_3 \\ 0 \end{bmatrix}$

$$J_{x|y} = J_x + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

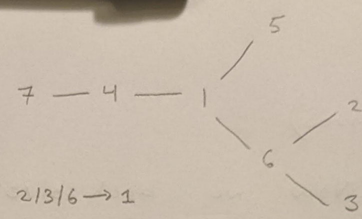
Thus

$$\boxed{\begin{aligned} h_{x_3 \to x_2} &= -J_{x_2 x_3} (J_{x_3 x_3})^{-1} h_{x_3} \\ &= -(J_x[2,3])(J_x[3,3] + 2)^{-1}(h_x[3] + y_2 + y_3) \\ J_{x_3 \to x_2} &= -J_{x_2 x_3} (J_{x_3 x_3})^{-1} J_{x_3 x_2} \\ &= -(J_x[2,3])(J_x[3,3] + 2)^{-1}(J_x[3,2]) \end{aligned}}$$

3b

3.3

b) The given matrix yields the graph



Order $7\to4$, $2\to6$, $2/3\to6$, $5\to1$, $4/7\to1$, $2/3/6\to1$

$$6(\text{row }4)-(\text{row }7) = \begin{bmatrix}6 & 0 & 0 & 18 & 0 & 0 & 6\end{bmatrix} - \begin{bmatrix}0 & 0 & 0 & -3 & 0 & 0 & 6\end{bmatrix} = \begin{bmatrix}6 & 0 & 0 & 21 & 0 & 0 & 0\end{bmatrix}$$

$6(24) - 12 = 132$ $\boxed{\text{row } 7\to4}$

$$4(\text{row }6)+(\text{row }2) = \begin{bmatrix}4 & -4 & 4 & 0 & 0 & 20 & 0\end{bmatrix} + \begin{bmatrix}0 & 4 & 0 & 0 & 0 & 1 & 0\end{bmatrix} = \begin{bmatrix}4 & 0 & -4 & 0 & 0 & 21 & 0\end{bmatrix}$$

$4(12) + 32 = 80$ $\boxed{\text{row } 2\to6}$

$$(\text{row }2\to6)+2(\text{row }3) = \begin{bmatrix}4 & 0 & -4 & 0 & 0 & 21 & 0\end{bmatrix} + \begin{bmatrix}0 & 0 & 4 & 0 & 0 & 2 & 0\end{bmatrix} = \begin{bmatrix}4 & 0 & 0 & 0 & 0 & 23 & 0\end{bmatrix}$$

$80 + 2(8) = 96$ $\boxed{\text{row } 2/3\to6}$

$$(\text{row }1)-(\text{row }5) = \begin{bmatrix}1 & 0 & 0 & -4 & 1 & -3 & 0\end{bmatrix} - \begin{bmatrix}2 & 0 & 0 & 0 & 1 & 0 & 0\end{bmatrix} = -1 \;\; 0 \;\; 0 \;\; -4 \;\; 0 \;\; -3 \;\; 0$$

$-32 - 5 = -37$ $\boxed{\text{row } 5\to1}$

$$21(\text{row }5\to1)+4(\text{row }7\to4) = \begin{bmatrix}-21 & 0 & 0 & -84 & 0 & -63 & 0\end{bmatrix} + \begin{bmatrix}24 & 0 & 0 & 84 & 0 & 0 & 0\end{bmatrix} = \begin{bmatrix}3 & 0 & 0 & 0 & 0 & -63 & 0\end{bmatrix}$$

$(21)(-37) + (4)(132) = -249$ $\boxed{\text{row } 4/5\to1}$

$$63(\text{row }2/3\to6)+23(\text{row }4/5\to1) = \begin{bmatrix}252 & 0 & 0 & 0 & 0 & -1449 & 0\end{bmatrix} + \begin{bmatrix}69 & 0 & 0 & 0 & 0 & 1449 & 0\end{bmatrix} = \begin{bmatrix}321 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$63(96) + 23(-249) = 321$ $\boxed{\text{row } 4/5/6\to1}$

3.3

b continued)

Now we back solve:

$$321 x_1 = 321 \Rightarrow \boxed{x_1 = 1}$$

$$3x_1 - 63x_6 = -249 \Rightarrow \boxed{x_6 = 4}$$

$$-x_1 - 4x_4 - 3x_6 = -37 \Rightarrow \boxed{x_4 = 6}$$

$$-4x_1 - 4x_3 + 21x_6 = 80 \Rightarrow \boxed{x_3 = 2}$$

$$-3x_4 + 6x_7 = 12 \Rightarrow \boxed{x_7 = 5}$$

$$4x_2 + x_6 = 32 \Rightarrow \boxed{x_2 = 7}$$

$$2x_1 + x_5 = 5 \Rightarrow \boxed{x_5 = 3}$$

Gaussian Elimination proceeds as Belief Propagation, working from leaves $\rightarrow$ roots and then back out again

Problem 4

4ab

3.4 Assume $1, 2, \ldots, N$ is a perfect elimination ordering

(a) Let $C_i = \{i\} \cup \{j \in N(i) : j > i\}$

C_i must be a clique b/c $\{j \in N(i) : j > i\}$ are the remaining nodes in the elimination ordering & eliminating i will connect all those neighbors to one another. Since elimination ordering is perfect, edges must already exist

$\Rightarrow \{i\} \cup \{j \in N(i) : j > i\}$ is a clique

(b) Let Ψ be a clique in G. We can write Ψ as

$$\Psi = \{m\} \cup S$$

where S is a possibly empty set of vertices s.t. $\forall i \in S, i > m$.

B/c Ψ is a clique, $\forall i \in S$, $\exists$ edge $(m, i) \in \mathcal{E}$ that also exists in G. Since $i \in N(i)$ and $i > m$, then

$\forall i \in S \Rightarrow i \in C_m$. Also, $m \in C_m$. Thus

$$\Psi \subseteq C_m$$

4c

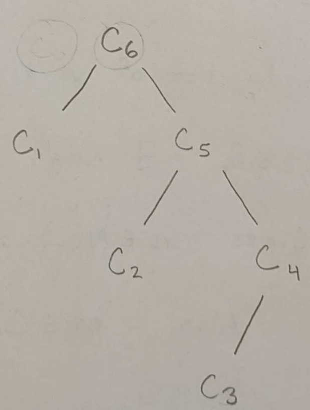
3.4

c)

i	C_i	$p(i)$	$C_{p(i)}$
1	$C_1 = \{1, 6\}$	6	C_6
2	$C_2 = \{2, 6, 5\}$	5	C_5
3	$C_3 = \{3, 4, 5\}$	4	C_4
4	$C_4 = \{4, 5, 6\}$	5	C_5
5	$C_5 = \{5, 6\}$	6	C_6
6	$C_6 = \{6\}$	$\emptyset$	$\emptyset$

Note: tabular form ↑ developed w/ Devin, Aviv, Anna & Ben

The rooted tree is therefore



4d

3.4)

d) Goal: Show that $C_i \setminus \{i\} \subseteq C_{p(i)}$

Using defn of C_i, $C_{p(i)}$, this is equivalent to showing

$$\{i\} \cup \{j \in N(i) : j > i\} \setminus \{i\} \subseteq C_{\min_{k \in C_i \setminus \{i\}} k}$$

$$\{j \in N(i) : j > i\} \subseteq C_{\min_{k \in C_i \setminus \{i\}} k}$$

Recall that the elimination ordering is a perfect ordering

$\Rightarrow$ no new edges introduced $\Rightarrow$ $\{j \in N(i) : j > i\}$ are

all pairwise connected

Since $\min_{k \in C_i \setminus \{i\}} k \in \{j \in N(i) : j > i\}$, these two facts

imply that $\{j \in N(i) : j > i\} \subseteq C_{\min_{k \in C_i \setminus \{i\}} k}$

Intuitively, since no edges are introduced, the smallest-index neighbor of i is also at least connected to all of i's neighbors

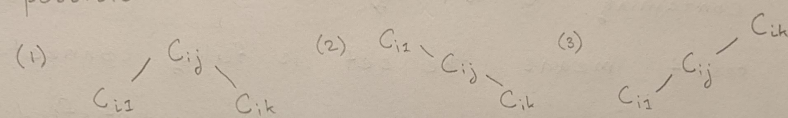
4e

3.4

e) I'll prove that $l \in C_{ij}$ $\forall j \in \{1, \dots, k\}$

Approach: for contradiction, assume $\exists C_{ij}$ s.t. $l \notin C_{ij}$ but $l \in C_{i1}$ and $l \in C_{ik}$

Base case: Conside $C_{i1} - C_{ij} - C_{ik}$. 3 arrangements are possible:



Case (1): By 3.4d, $(C_{i1} \setminus \{i1\}) \cup (C_{ik} \setminus \{ik\}) \subseteq C_{ij}$

Since $i1 \neq ik$, j must exist in at least $C_{i1} \setminus \{i1\}$ or $C_{ik} \setminus \{ik\}$ $\Rightarrow l \in C_{ij} \Rightarrow$ contradiction!

Case (2). By 3.4d, $(C_{ik} \setminus \{ik\}) \subseteq C_{ij}$ and $C_{ij} \setminus \{ij\} \subseteq C_{i1}$

Since $l \in C_{ik}$ and $l \in C_{i1}$, if $l \neq ik$, then $l \in C_{ij} \rightarrow$ contradiction! But if $l = ik$, then $l \notin C_1 \Rightarrow$ Contradiction!

Case (3) Same as Case (2) by symmetry

Now consider $C_{i1} - C_{i2} - \dots C_{ij} \dots - C_{ik}$. For contradiction, assume all nodes on path have l as an element, except C_{ij}. Then this simplifies to $C_{ij-1} - C_{ij} - C_{ij+1}$, the same as the base case. If not all have l, then we have a contradiction by the same reasoning above.

Conclusion: If $l \in C_{i1} \cap C_{ik}$, then $l \in C_{ij}$ for j on path

4fg

3.4

f) Identify all cliques containing ℓ. For each pair of these cliques $(C_{\ell 1}, C_{\ell k})$, we have $\ell \in C_{\ell 1} \cap C_{\ell k}$. From my 3.4e, I showed every clique on path connecting $C_{\ell 1}$ to $C_{\ell k}$ contains ℓ. The set of all cliques containing ℓ is by defn a subtree, and my path result means the set is fully connected. Consequently, the junction tree property is satisfied.

g)

Chordalization: $O(ND^2)$

Create the junction tree: $O(ND)$

Belief Prop. on tree: $O(N4^D)$

Total time complexity: $O(ND^2 + ND + N4^D)$

$= O(N4^D)$

Problem 5

5a & 5b

3.5

$$p(x) \propto 1 - \prod_{i=1}^{n} x_i$$

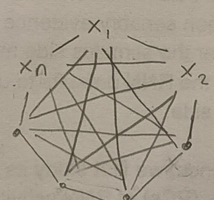
a) Recall that minimal I-map $\stackrel{\text{def}}{=}$ I-map + can't remove edges

I-map means $p(x)$ has at least CIs implied by graph

$p(x)$ has $\forall\, i,j$ $\; x_i \perp\!\!\!\perp x_j$ but no others e.g.

$$x_i \perp\!\!\!\perp x_j \mid x_k$$

b/c if $x_k = 1$, conditioning changes probabilities of x_i, x_j



← fully connected

b) Note that $KL(q\|p) = \sum_{\underline{x}} q(\underline{x}) \log \dfrac{q(\underline{x})}{p(\underline{x})}$ will be ∞ if

$q(\underline{1}) \neq 0$ b/c $q(\underline{1}) \log \dfrac{q(\underline{1})}{p(\underline{1})} = q(\underline{1}) \log \dfrac{q(\underline{1})}{0} = \infty$

$$\Rightarrow 0 = q(\underline{1}) = \prod_{i=1}^{n} f(1) = (f(1))^n \Rightarrow f(1) = 0 \Rightarrow f(0) = 1$$

Unique minimizer $\boxed{q(\underline{x}) = \prod_{i=1}^{n} f(x_i) \text{ where } f(0) = 1,\ f(1) = 0}$

5c

3.5 c)

Let $q(\underline{x}) \overset{def}{=} \prod_{n=1}^{N} q_n(x_n)$ and $Z = 2^N - 1$

Fix $q_1(x_1)$ to $q_1(1) = 0$ and $q_1(0) = 1$. Then KL:

$$KL(q \| p) = \sum_{\underline{x}} q(\underline{x}) \log \frac{q(\underline{x})}{p(\underline{x})}$$

$$= \sum_{\underline{x} : x_1 = 0} q(\underline{x}) \log \frac{q(\underline{x})}{p(\underline{x})}$$

$$= \underbrace{\sum_{\underline{x} : x_1 = 0} q(\underline{x}) \log q(\underline{x})}_{-H(x)} - \underbrace{\sum_{\underline{x} : x_1 = 0} q(\underline{x}) \cancelto{-\log Z}{\log p(\underline{x})}}_{+ \log Z \sum_{\underline{x} : x_1 = 0} \cancelto{1}{q(\underline{x})}}$$

Minimizing KL $\iff$ Minimize $-H(x)$ $\iff$ Maximize $H(x)$

$$\Rightarrow \forall\, n \in \{1 \dots N\} \text{ s.t. } n \neq 1, \quad q_n(x_n) = \begin{cases} 1/2 & x_n = 0 \\ 1/2 & x_n = 1 \end{cases}$$

This is $\underline{1}$ soln, w/ $q_1(x_1) = \begin{cases} 1 & x_n = 0 \\ 0 & x_n = 1 \end{cases}$

We could have chosen any $q_i(x_i) = \begin{cases} 1 & x_i = 0 \\ 0 & x_i = 1 \end{cases}$ for any $i \in \{1 \dots N\}$

$\Rightarrow$ There are N symmetric solns s.t $\forall\, i \in \{1 \dots N\}$

$$q_i(x_i) = \begin{cases} 1 & x_i = 0 \\ 0 & x_i = 1 \end{cases} \quad \text{and} \quad q_{\text{not } i}(x_{\text{not } i}) = \begin{cases} 1/2 & x_{\text{not } i} = 0 \\ 1/2 & x_{\text{not } i} = 1 \end{cases}$$

is a valid minimizer of $KL(q \| p)$

Problem 6

6a

3.6

a) Consider factorized variational distribution $q(\underline{x}) = \prod_{n=1}^{N} q_n(x_n)$

$$\min_{q_k(x_k)} KL(p\|q) = \min_{q_k} \sum_{\underline{x}} p(\underline{x}) \log \frac{p(\underline{x})}{q(\underline{x})}$$

$$= \min_{q_k} \underbrace{\sum_{\underline{x}} p(\underline{x}) \log p(\underline{x})}_{\text{doesn't depend on } q(\underline{x})} \underbrace{- \sum_{\underline{x}} p(\underline{x}) \log q(\underline{x})}_{CE(p\|q)}$$

Equivalently, we can minimize cross entropy:

$$\min_{q_k(x_k)} - \sum_{\underline{x}} p(\underline{x}) \log q(\underline{x}) = \min_{q_k(x_k)} - \sum_{\underline{x}} p(\underline{x}) \sum_{n=1}^{N} \log q_n(x_n)$$

$$= \min_{q_k(x_k)} - \sum_{n=1}^{N} \sum_{\underline{x}} p(\underline{x}) \log q_n(x_n)$$

$$= \min_{q_k(x_k)} - \sum_{n=1}^{N} p_n(x_n) \log q_n(x_n)$$

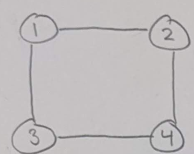
$\Rightarrow$ Need to minimize CE between $p_k(x_k)$ and $q_k(x_k)$ $\forall k \in \{1 \ldots N\}$

Since marginal $p_k(x_k)$ is intractable, this minimization problem is also intractable

6b

3.6

b) $q_x(\underline{x}) = \prod_{(i,j)\in\mathcal{E}_b} q_{ij}(x_i, x_j)$



Suppose

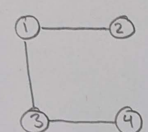
$\psi_{12} = \mathbb{1}_{x_1 = x_2}$

$\psi_{34} = \mathbb{1}_{x_3 = x_4}$

$\psi_{24} = \mathbb{1}_{x_2 = x_4}$

$\psi_{13} = 1 + x_1 + x_3$

Now consider



Compare $\quad q(\underline{x}) \propto \psi_{12}\,\psi_{13}\,\psi_{34}$

$$\Rightarrow KL(q\|p) = \sum_{\underline{x}} q(\underline{x}) \log \frac{q(\underline{x})}{p(x)}$$

$$= \sum_{\underline{x}} \psi_{12}\psi_{13}\psi_{34} \log \frac{\psi_{12}\psi_{13}\psi_{34}}{\psi_{12}\psi_{13}\psi_{34}\psi_{24}}$$

$= \infty \quad$ b/c $\quad p(1,1,0,0) = 0$ but

$q(1,1,0,0) > 0 \Rightarrow KL = \infty$

with $\quad q(\underline{x}) = \psi_{12}\,\mathbb{1}_{1=3}\,\psi_{34}$

$$\Rightarrow KL(q\|p) = \sum \mathbb{1}_{1=2}\,\mathbb{1}_{1=3}\,\mathbb{1}_{3=4} \log \frac{\mathbb{1}_{1=2}\,\mathbb{1}_{1=3}\,\mathbb{1}_{3=4}}{\mathbb{1}_{1=2}(1+x_1+x_3)\mathbb{1}_{3=4}\,\mathbb{1}_{24}}$$

$< \infty$

$\boxed{\text{FALSE}}$

Intuitively, if $q(\underline{x})$ matches edge potentials in $\mathcal{E}_B$, it might have disjoint support from $p(x)$

6c

3.6

c)

Consider graph I:

$$D_{KL}(q^{I} \| p) = \sum_{\underline{x}} q^{I}(\underline{x}) \log \frac{q^{I}(\underline{x})}{p(\underline{x})}$$

$$= \sum_{\underline{x}} q_{14}^{I} q_{23}^{I} \left[\log q_{14}^{I} + \log q_{23}^{I} - \log p(\underline{x}) \right]$$

$$= \sum_{x_1 x_4} q_{14}^{I} \log q_{14}^{I} \cancelto{1}{\sum_{x_2 x_3} q_{23}^{I}} + \sum_{x_2 x_3} q_{23}^{I} \log q_{23}^{I} \cancelto{1}{\sum_{x_1 x_4} q_{14}^{I}}$$

$$- \sum_{\underline{x}} q_{14}^{I} q_{23}^{I} \log p(x)$$

$$\partial_{q_{14}^{I}(x_1,x_4)} D_{KL} = 0 = \log q_{14}^{I}(x_1,x_4) + 1 - \sum_{\substack{x_2 x_3 \\ x_1 = x_1 \\ x_4 = x_4}} q_{23}^{I} \log \frac{\psi_{12}\psi_{24}\psi_{34}\psi_{13}}{Z}$$

$$\Rightarrow q_{14}^{I}(x_1, x_4) = \exp\left(-1 + \sum_{\substack{x_2 x_3 \\ x_1 = x_1 \\ x_4 = x_4}} q_{23}^{I} \log \frac{\psi_{12}\psi_{24}\psi_{34}\psi_{13}}{Z}\right)$$

$$= \exp\left(-1 + \sum_{\substack{x_2 x_3 \\ x_1 = 1}} q_{23}^{I} \log \frac{\psi_{12}\psi_{13}}{Z}\right) \exp\left(\sum_{\substack{x_2 x_3 \\ x_4 = x_4}} q_{23}^{I} \log \psi_{34}\psi_{24}\right)$$

$$= q_1^{I}(x_1)\, q_4^{I}(x_1)$$

$\Rightarrow$ Updates for Graph I same as updates for Graph II

$$\Rightarrow \boxed{\text{Graph I} = \text{Graph II}}$$

(also holds for $q_{23}^{I}(x_2, x_3) = q_2^{I}(x_2)\, q_3^{I}(x_3)$ by symmetry)

$\underline{3.6}$

c con't) Look at $\underline{III}$

$$KL(q^{III} \| p) = \sum_{\underline{x}} q^{III}(\underline{x}) \log \frac{q^{III}(x)}{p(x)}$$

$$= \sum_{\underline{x}} q_{12}^{III} q_{23}^{III} q_{34}^{III} \log \frac{q_{12}^{III} q_{23}^{III} q_{34}^{III}}{p(\underline{x})}$$

$$= \sum_{x_1 x_2} q_{12}^{III} \log q_{12}^{III} + \sum_{x_2 x_3} q_{23}^{III} \log q_{23}^{III} + \sum_{x_3 x_4} q_{34}^{III} \log q_{34}^{III}$$

$$- \sum_{x} q_{12}^{III} q_{23}^{III} q_{34}^{III} \log p(\underline{x})$$

$$\partial_{q_{23}(x_2, x_3)} KL(q^{III} \| p) = 0 + \log q_{23}^{III}(x_2, x_3) + 1 + 0 - \sum_{\substack{x_1 x_4 \\ x_2 = x_2 \\ x_3 = x_3}} q_{12}^{III} q_{34}^{III} \log p(x)$$

$$\Rightarrow q_{23}^{III} = \exp\Big(-1 + \sum_{\substack{x_1 x_4 \\ x_2 = x_2 \\ x_3 = x_3}} q_{12} q_{34} \big[\log \psi_{12} + \log \psi_{24} + \log \psi_{34} + \log \psi_{13}\big]\Big)$$

$$\Rightarrow \cancel{q_{23}^{III} = \exp\Big(-1 + \sum q_{12} q_{34} \big[\log \psi_{12} + \log \psi_{24}\big]\Big) \exp\Big(\sum q_{12} q_{34} \big[\log \psi_{34} + \log \psi_{13}\big]\Big)}$$

Can't split because $q_{12} q_{34}$ is fn. of both 2 and 3

$$\Rightarrow \boxed{\text{Graph III} \geq \text{IV}}$$

Final answer $\boxed{III \geq IV \geq II = I}$

Computational

Cai

IMPORTANT: My nodes are ordered using a DFS, prioritizing left first i.e. Orville is 0, Abraham is 1, Homer is 2, Lisa is 3, etc.

All marginals when all nodes are observed are uniform i.e. 0.5/0.5.

```
Marginals After Conditioning (alpha=2, beta=1.0)
[[ 0.      0.515  0.485]
 [ 1.      0.455  0.545]
 [ 2.      0.337  0.663]
 [ 3.      1.     0.   ]
 [ 4.      0.     1.   ]
 [ 5.      0.     1.   ]
 [ 6.      0.594  0.406]
 [ 7.      0.799  0.201]
 [ 8.      1.     0.   ]
 [ 9.      1.     0.   ]
 [10.      0.531  0.469]]
 Marginals After Conditioning (alpha=4, beta=1.0)
[[ 0.      0.555  0.445]
 [ 1.      0.409  0.591]
 [ 2.      0.239  0.761]
 [ 3.      1.     0.   ]
 [ 4.      0.     1.   ]
 [ 5.      0.     1.   ]
 [ 6.      0.715  0.285]
 [ 7.      0.932  0.068]
 [ 8.      1.     0.   ]
 [ 9.      1.     0.   ]
 [10.      0.629  0.371]]
 Marginals After Conditioning (alpha=6, beta=1.0)
[[ 0.      0.572  0.428]
 [ 1.      0.4    0.6  ]
 [ 2.      0.216  0.784]
 [ 3.      1.     0.   ]
 [ 4.      0.     1.   ]
 [ 5.      0.     1.   ]
 [ 6.      0.752  0.248]
 [ 7.      0.961  0.039]
 [ 8.      1.     0.   ]
 [ 9.      1.     0.   ]
 [10.      0.68   0.32 ]]
 Marginals After Conditioning (alpha=8, beta=1.0)
[[ 0.      0.58   0.42 ]
 [ 1.      0.397  0.603]
 [ 2.      0.208  0.792]
 [ 3.      1.     0.   ]
 [ 4.      0.     1.   ]
 [ 5.      0.     1.   ]
 [ 6.      0.768  0.232]
 [ 7.      0.973  0.027]
 [ 8.      1.     0.   ]
 [ 9.      1.     0.   ]
```

```
 [10.        0.708   0.292]]
 Marginals  After  Conditioning  (alpha=10,  beta=1.0)
[[  0.        0.585   0.415]
 [  1.        0.397   0.603]
 [  2.        0.204   0.796]
 [  3.        1.      0.   ]
 [  4.        0.      1.   ]
 [  5.        0.      1.   ]
 [  6.        0.776   0.224]
 [  7.        0.979   0.021]
 [  8.        1.      0.   ]
 [  9.        1.      0.   ]
 [10.        0.726   0.274]]
```

Caii

IMPORTANT: My nodes are ordered using a DFS, prioritizing left first i.e. Orville is 0, Abraham is 1, Homer is 2, Lisa is 3, etc.

```
Marginals  After  Conditioning  (alpha=2,  beta=0.8)
[[  0.        0.51    0.49 ]
 [  1.        0.474   0.526]
 [  2.        0.402   0.598]
 [  3.        0.756   0.244]
 [  4.        0.201   0.799]
 [  5.        0.201   0.799]
 [  6.        0.56    0.44 ]
 [  7.        0.691   0.309]
 [  8.        0.82    0.18 ]
 [  9.        0.82    0.18 ]
 [10.        0.52    0.48 ]]
Marginals  After  Conditioning  (alpha=2,  beta=0.9)
[[  0.        0.513   0.487]
 [  1.        0.465   0.535]
 [  2.        0.37    0.63 ]
 [  3.        0.866   0.134]
 [  4.        0.1     0.9  ]
 [  5.        0.1     0.9  ]
 [  6.        0.578   0.422]
 [  7.        0.748   0.252]
 [  8.        0.915   0.085]
 [  9.        0.915   0.085]
 [10.        0.526   0.474]]
Marginals  After  Conditioning  (alpha=4,  beta=0.8)
[[  0.        0.551   0.449]
 [  1.        0.46    0.54 ]
 [  2.        0.357   0.643]
 [  3.        0.658   0.342]
 [  4.        0.216   0.784]
 [  5.        0.216   0.784]
 [  6.        0.657   0.343]
 [  7.        0.804   0.196]
 [  8.        0.855   0.145]
 [  9.        0.855   0.145]
```

```
 [10.        0.594   0.406]]
Marginals  After  Conditioning  (alpha=4,  beta=0.9)
[[  0.        0.557   0.443]
 [  1.        0.437   0.563]
 [  2.        0.301   0.699]
 [  3.        0.777   0.223]
 [  4.        0.111   0.889]
 [  5.        0.111   0.889]
 [  6.        0.692   0.308]
 [  7.        0.877   0.123]
 [  8.        0.939   0.061]
 [  9.        0.939   0.061]
 [10.        0.615   0.385]]
Marginals  After  Conditioning  (alpha=6,  beta=0.8)
[[  0.        0.582   0.418]
 [  1.        0.477   0.523]
 [  2.        0.369   0.631]
 [  3.        0.607   0.393]
 [  4.        0.247   0.753]
 [  5.        0.247   0.753]
 [  6.        0.696   0.304]
 [  7.        0.833   0.167]
 [  8.        0.867   0.133]
 [  9.        0.867   0.133]
 [10.        0.64    0.36  ]]
Marginals  After  Conditioning  (alpha=6,  beta=0.9)
[[  0.        0.585   0.415]
 [  1.        0.446   0.554]
 [  2.        0.301   0.699]
 [  3.        0.715   0.285]
 [  4.        0.133   0.867]
 [  5.        0.133   0.867]
 [  6.        0.733   0.267]
 [  7.        0.909   0.091]
 [  8.        0.947   0.053]
 [  9.        0.947   0.053]
 [10.        0.667   0.333]]
Marginals  After  Conditioning  (alpha=8,  beta=0.8)
[[  0.        0.604   0.396]
 [  1.        0.499   0.501]
 [  2.        0.393   0.607]
 [  3.        0.583   0.417]
 [  4.        0.28    0.72  ]
 [  5.        0.28    0.72  ]
 [  6.        0.717   0.283]
 [  7.        0.843   0.157]
 [  8.        0.87    0.13  ]
 [  9.        0.87    0.13  ]
 [10.        0.668   0.332]]
Marginals  After  Conditioning  (alpha=8,  beta=0.9)
[[  0.        0.604   0.396]
 [  1.        0.461   0.539]
 [  2.        0.315   0.685]
 [  3.        0.673   0.327]
```

```
 [ 4.        0.158   0.842]
 [ 5.        0.158   0.842]
 [ 6.        0.754   0.246]
 [ 7.        0.921   0.079]
 [ 8.        0.95    0.05  ]
 [ 9.        0.95    0.05  ]
 [10.        0.698   0.302]]
Marginals  After  Conditioning  (alpha=10,  beta=0.8)
[[ 0.        0.622   0.378]
 [ 1.        0.519   0.481]
 [ 2.        0.417   0.583]
 [ 3.        0.574   0.426]
 [ 4.        0.312   0.688]
 [ 5.        0.312   0.688]
 [ 6.        0.729   0.271]
 [ 7.        0.846   0.154]
 [ 8.        0.869   0.131]
 [ 9.        0.869   0.131]
 [10.        0.687   0.313]]
Marginals  After  Conditioning  (alpha=10,  beta=0.9)
[[ 0.        0.62    0.38  ]
 [ 1.        0.477   0.523]
 [ 2.        0.333   0.667]
 [ 3.        0.645   0.355]
 [ 4.        0.183   0.817]
 [ 5.        0.183   0.817]
 [ 6.        0.768   0.232]
 [ 7.        0.926   0.074]
 [ 8.        0.951   0.049]
 [ 9.        0.951   0.049]
 [10.        0.719   0.281]]
```

All marginals when all nodes are observed are uniform i.e. 0.5/0.5.

Cbi

All marginals when all nodes are observed are uniform i.e. 0.5/0.5.

```
Marginals  After  Conditioning  (alpha=2,  beta=1.0)
[[   0.        0.2       0.8    ]
 [  50.        0.231     0.769]
 [100.        0.37      0.63   ]
 [150.        0.41      0.59   ]
 [200.        0.334     0.666]
 [250.        0.5       0.5    ]
 [300.        0.2       0.8    ]
 [350.        0.6       0.4    ]]
Marginals  After  Conditioning  (alpha=4,  beta=1.0)
[[   0.        0.059     0.941]
 [  50.        0.012     0.988]
 [100.        0.118     0.882]
 [150.        0.207     0.793]
 [200.        0.123     0.877]
 [250.        0.5       0.5    ]
 [300.        0.059     0.941]
```

```
 [350.        0.765    0.235]]
Marginals After Conditioning (alpha=6, beta=1.0)
[[  0.        0.027    0.973]
 [ 50.        0.001    0.999]
 [100.        0.055    0.945]
 [150.        0.144    0.856]
 [200.        0.058    0.942]
 [250.        0.5      0.5  ]
 [300.        0.027    0.973]
 [350.        0.838    0.162]]
Marginals After Conditioning (alpha=8, beta=1.0)
[[  0.        0.015    0.985]
 [ 50.        0.       1.   ]
 [100.        0.031    0.969]
 [150.        0.111    0.889]
 [200.        0.033    0.967]
 [250.        0.5      0.5  ]
 [300.        0.015    0.985]
 [350.        0.877    0.123]]
```

Cbii

All marginals when all nodes are observed are uniform i.e. 0.5/0.5.

```
Computational b.ii
Marginals After Conditioning (alpha=2, beta=0.8)
[[  0.        0.314    0.686]
 [ 50.        0.204    0.796]
 [100.        0.379    0.621]
 [150.        0.401    0.599]
 [200.        0.392    0.608]
 [250.        0.488    0.512]
 [300.        0.316    0.684]
 [350.        0.565    0.435]]
Marginals After Conditioning (alpha=2, beta=0.9)
[[  0.        0.256    0.744]
 [ 50.        0.19     0.81 ]
 [100.        0.365    0.635]
 [150.        0.397    0.603]
 [200.        0.361    0.639]
 [250.        0.49     0.51 ]
 [300.        0.256    0.744]
 [350.        0.583    0.417]]
Marginals After Conditioning (alpha=4, beta=0.8)
[[  0.        0.219    0.781]
 [ 50.        0.009    0.991]
 [100.        0.174    0.826]
 [150.        0.205    0.795]
 [200.        0.251    0.749]
 [250.        0.443    0.557]
 [300.        0.227    0.773]
 [350.        0.7      0.3  ]]
Marginals After Conditioning (alpha=4, beta=0.9)
[[  0.        0.132    0.868]
```

```
 [ 50.        0.005    0.995]
 [100.        0.142    0.858]
 [150.        0.203    0.797]
 [200.        0.184    0.816]
 [250.        0.449    0.551]
 [300.        0.135    0.865]
 [350.        0.737    0.263]]
Marginals After Conditioning (alpha=6, beta=0.8)
[[  0.        0.205    0.795]
 [ 50.        0.001    0.999]
 [100.        0.153    0.847]
 [150.        0.144    0.856]
 [200.        0.264    0.736]
 [250.        0.417    0.583]
 [300.        0.216    0.784]
 [350.        0.779    0.221]]
Marginals After Conditioning (alpha=6, beta=0.9)
[[  0.        0.104    0.896]
 [ 50.        0.       1.   ]
 [100.        0.104    0.896]
 [150.        0.143    0.857]
 [200.        0.162    0.838]
 [250.        0.428    0.572]
 [300.        0.11     0.89 ]
 [350.        0.814    0.186]]
Marginals After Conditioning (alpha=8, beta=0.8)
[[  0.        0.21     0.79 ]
 [ 50.        0.       1.   ]
 [100.        0.176    0.824]
 [150.        0.111    0.889]
 [200.        0.332    0.668]
 [250.        0.409    0.591]
 [300.        0.222    0.778]
 [350.        0.829    0.171]]
 Marginals After Conditioning (alpha=8, beta=0.9)
[[  0.        0.096    0.904]
 [ 50.        0.       1.   ]
 [100.        0.108    0.892]
 [150.        0.111    0.889]
 [200.        0.191    0.809]
 [250.        0.42     0.58 ]
 [300.        0.104    0.896]
 [350.        0.857    0.143]]
```

Code

```python
import numpy as np

np.set_printoptions(3, suppress=True)


def parse_tree_txt(file_path):
    with open(file_path, 'r') as fp:
```

```python
        n = int(fp.readline())
        nodes = np.arange(n)
        edges = {}
        for i in range(n):
            edges[i] = set()
        traits = np.ones(shape=(n, 2))
        for line in fp.readlines():
            part1, part2 = line[:-1].split(' ')
            if part2.isdigit():
                edges[int(part1)].add(int(part2))
                edges[int(part2)].add(int(part1))
            else:
                if part2 == 'A':
                    traits[int(part1), 1] = 0.
                elif part2 == 'B':
                    traits[int(part1), 0] = 0.

    return nodes, edges, traits


def compute_marginals(nodes,
                      edges,
                      node_potentials,
                      edge_potentials):

    # initialize messages with keys (from, to)
    messages = {}
    for j_node in edges:
        for i_node in edges[j_node]:
            messages[(i_node, j_node)] = np.ones(shape=2)

    # compute messages
    for _ in range(5 * len(edges)):
        for j_node in edges:
            for i_node in edges[j_node]:
                i_neighbors = edges[i_node] - set([j_node])
                message_product = np.ones(shape=2)
                for i_neighbor in i_neighbors:
                    incoming_message = messages[(i_neighbor, i_node)]
                    message_product = np.multiply(
                        message_product,
                        incoming_message)
                node_times_message_product = np.multiply(
                    node_potentials[i_node],
                    message_product)
                edge_times_node_times_message_product = np.matmul(
                    edge_potentials,
                    node_times_message_product)
                messages[(i_node, j_node)] = edge_times_node_times_message_product
                messages[(i_node, j_node)] /= np.sum(messages[(i_node, j_node)])

    # compute marginals
    marginals = np.zeros(shape=(len(nodes), 2))
    for node in nodes:
```

```
        message_product = np.ones(shape=2)
        node_neighbors = edges[node]
        for node_neighbor in node_neighbors:
            incoming_message = messages[(node_neighbor, node)]
            message_product = np.multiply(
                message_product,
                incoming_message)
        node_times_message_product = np.multiply(
            node_potentials[node],
            message_product)
        marginal = node_times_message_product / np.sum(node_times_message_product)
        marginals[node] = marginal

    return marginals


def compute_marginals_before_and_after(tree_path,
                                       alphas,
                                       betas,
                                       part,
                                       indices_to_print=None):
    print(part)
    nodes, edges, traits = parse_tree_txt(tree_path)
    for alpha in alphas:
        for beta in betas:

            # before conditioning
            node_potentials = np.ones(shape=(len(nodes), 2))
            edge_potentials = np.array([[alpha, 1], [1, alpha]])
            marginals = compute_marginals(nodes=nodes,
                                          edges=edges,
                                          node_potentials=node_potentials,
                                          edge_potentials=edge_potentials)
            print(f'Marginals Before Conditioning (alpha={alpha}, beta={beta})')
            if indices_to_print is not None:
                print(np.concatenate([np.expand_dims(nodes, axis=1),
                                      marginals], axis=1)[indices_to_print])
            else:
                print(np.concatenate([np.expand_dims(nodes, axis=1), marginals], axis=1))

            # after conditioning
            # first update node potentials
            observed_rows = traits.sum(axis=1) == 1
            node_potentials[observed_rows] = beta * traits[observed_rows] + \
                                             (1 - beta) * np.abs(1 - traits[observed_rows]
            # node_potentials = np.multiply(node_potentials, traits)
            node_potentials = np.divide(node_potentials,
                                        np.sum(node_potentials,
                                               axis=1,
                                               keepdims=True))
            conditional_marginals = compute_marginals(nodes=nodes,
                                                      edges=edges,
                                                      node_potentials=node_potentials,
                                                      edge_potentials=edge_potentials)
```

```python
            print(f'Marginals After Conditioning (alpha={alpha}, beta={beta})')
            if indices_to_print is not None:
                print(np.concatenate([np.expand_dims(nodes, axis=1),
                                      conditional_marginals],
                                     axis=1)[indices_to_print])
            else:
                print(np.concatenate([np.expand_dims(nodes, axis=1),
                                      conditional_marginals],
                                     axis=1))


# part a.i
alphas = np.arange(2, 11, 2)
betas = np.array([1.])
compute_marginals_before_and_after(
    tree_path='/home/rylan/Documents/MIT6.438-Probabilistic-Graphical-Models/hw3/toy-tree.t
    alphas=alphas,
    betas=betas,
    part='Computational a.i',
)

# part a.ii
alphas = np.arange(2, 11, 2)
betas = np.array([0.8, 0.9])
compute_marginals_before_and_after(
    tree_path='/home/rylan/Documents/MIT6.438-Probabilistic-Graphical-Models/hw3/toy-tree.t
    alphas=alphas,
    betas=betas,
    part='Computational a.ii',
)

# part b.i
alphas = np.arange(2, 9, 2)
betas = np.array([1.])
compute_marginals_before_and_after(
    tree_path='/home/rylan/Documents/MIT6.438-Probabilistic-Graphical-Models/hw3/family-tre
    alphas=alphas,
    betas=betas,
    part='Computational b.i',
    indices_to_print=np.arange(0, 351, 50),
)

# part b.ii
alphas = np.arange(2, 9, 2)
betas = np.array([0.8, 0.9])
compute_marginals_before_and_after(
    tree_path='/home/rylan/Documents/MIT6.438-Probabilistic-Graphical-Models/hw3/family-tre
    alphas=alphas,
    betas=betas,
    part='Computational b.ii',
    indices_to_print=np.arange(0, 351, 50),
)
```