6.1 Submitted

6.2

a) $$P(a,b|c,d) = \frac{P(a,b,c,d)}{P(c,d)}$$ $\boxed{P.R.}$

$$= \frac{P(a)P(b|a)P(c|a,b)P(d|a,b,c)}{P(c,d)}$$

$\boxed{P.R.}$

$$= \frac{P(a)P(b|a)P(c|a,b)P(d|b,c)}{\sum_a \sum_b P(a,b,c,d)}$$

$\boxed{\text{rule-1 \& marg}}$

$$= \frac{P(a)P(b|a)P(c|a,b)P(d|b,c)}{\sum_a \sum_b P(a)P(b|a)P(c|a,b)P(d|b,c)}$$

$\boxed{\text{similar to numerator}}$

b)

$$P(a|c,d) = \sum_b P(a,b|c,d)$$

$\boxed{\text{marg}}$

$$P(b|c,d) = \sum_a P(a,b|c,d)$$

$\boxed{\text{marg}}$

c) $$\mathcal{L} = \sum_t^T \log P(C=c_t, D=d_t)$$

$$= \sum_t^T \log \sum_a \sum_b P(A=a, B=b, C=c_t, D=d_t)$$

$\boxed{\text{marg}}$

$$= \sum_t^T \log \sum_a \sum_b P(A=a) P(B=b \mid A=a) P(C=c_t \mid A=a, B=b)$$

$$\cdot P(D=d_t \mid A=a, B=b, C=c_t)$$

$\boxed{\text{P.R.}}$

$$= \sum_t^T \log \sum_a \sum_b P(A) P(B \mid A) P(C \mid A, B) P(D \mid B, C)$$

$\boxed{\text{rule 1}}$

d) ① $$P(A=a) \leftarrow \frac{1}{T} \sum_t^T P(a \mid c_t, d_t)$$ $\boxed{\text{root node}}$

② $$P(B=b \mid A=a) \leftarrow \frac{\sum_t^T P(a, b \mid c_t, d_t)}{\sum_t^T P(a \mid c_t, d_t)}$$

③ $$P(C=c \mid A=a, B=b) \leftarrow \frac{\sum_t^T P(a, b, c \mid c_t, d_t)}{\sum_t^T P(a, b \mid c_t, d_t)}$$

$$\leftarrow \frac{\sum_t^T P(a, b \mid c_t, d_t)\, I(c, c_t)}{\sum_t^T P(a, b \mid c_t, d_t)}$$

4

$$P(D=d \mid B=b, C=c) \leftarrow \frac{\sum_{t}^{T} P(B=b, C=c, D=d \mid c_t, d_t)}{\sum_{t}^{T} P(B=b, C=c \mid c_t, d_t)}$$

$$\leftarrow \frac{\sum_{t}^{T} I(c, c_t) I(d, d_t) P(b \mid c_t, d_t)}{\sum_{t}^{T} I(c, c_t) P(b \mid c_t, d_t)}$$

6.3

a)

$$\sum_{Z \in \{0,1\}^n} P(Y=1, Z \mid X)$$

$$\Rightarrow \sum_{Z \in \{0,1\}^n} P(Y=1 \mid Z, X) P(Z \mid X) \qquad \boxed{P.R.}$$

$$\Rightarrow \sum_{Z \in \{0,1\}^n} P(Y=1 \mid Z) P(Z \mid X) \qquad \boxed{\text{Rule } 1}$$

$$\Rightarrow \sum_{Z \in \{0,1\}^n} (1 - I(Z, \{0\}^n)) P(Z \mid X)$$

$\because$ $P(Y=1 \mid Z) = \begin{cases} 1 & \text{if any } Z_i \text{ is } 1 \\ 0 & \text{if all } Z_i \text{ are } 0 \end{cases}$

$\Rightarrow P(Y=0 \mid Z) = \begin{cases} 0 & \text{if any } Z_i \text{ is } 1 \\ 1 & \text{if all } Z_i \text{ are } 0 \end{cases}$

$\Rightarrow$ when $Z == \{0\}^n$ $\Rightarrow P(Y=0 \mid Z=\{0\}^n) = 1$

$\Rightarrow P(Y=0 \mid Z \neq \{0\}^n) = 0$

$\Rightarrow 1 - P(Y=0 \mid Z=\{0\}^n) = P(Y=1 \mid Z=\{0\}^n) = 0$

$1 - P(Y=0 \mid Z \neq \{0\}^n) = P(Y=1 \mid Z \neq \{0\}^n) = 1$

$\Rightarrow 1 - I(Z, \{0\}^n) = P(Y=1 \mid Z)$

$$\Rightarrow \sum_{Z\in\{0,1\}^n} P(Z|X) - \sum_{Z\in\{0,1\}} P(Z|X)\, I(Z,\{0\}^n)$$

$$\Rightarrow 1 - \sum_{Z\in\{0,1\}} P(Z|X)\, I(Z,\{0\}^n)$$ marg

$$\Rightarrow 1 - P(Z=\{0\}^n | X)$$ Indicator f^n is zero for rest

$$\Rightarrow 1 - \prod_{i=1}^{n} P(Z_i = 0 | X)$$ rule-3

$$\Rightarrow 1 - \prod_{i=1}^{n} P(Z_i = 0 | X_i)$$ rule-3

$$\Rightarrow 1 - \prod_{i=1}^{n} (1-P_i)^{X_i}(1)^{1-X_i}$$

$$\Rightarrow \boxed{1 - \prod_{i=1} (1-P_i)^{X_i}}$$

b) $P(Z_i = 1, X_i = 1 \mid X = x, Y = y)$

$\Rightarrow \quad I(x_i, 1)\, P(Z_i = 1 \mid X, Y)$ $\qquad \boxed{X_i \text{ is observed node}}$

$\Rightarrow \quad x_i \left[\dfrac{P(Z_i = 1 \mid X)\, P(Y \mid X, Z_i = 1)}{P(Y \mid X)} \right]$ $\qquad \boxed{\text{Bayes}}$

$\Rightarrow \quad x_i \left[\dfrac{P(Z_i = 1 \mid X_i)\, P(Y \mid Z_i = 1)}{P(Y \mid X)} \right]$ $\qquad \boxed{\text{rule-1 \& rule-3}}$

$\Rightarrow \quad \because$ Y is the observed node

$\Rightarrow \quad x_i \left[\dfrac{p_i \; I(y_i, 1)}{P(Y \mid X)} \right]$

$\Rightarrow \quad \dfrac{x_i\, p_i\, y_i}{P(Y \mid X)}$

$\Rightarrow \quad \dfrac{x_i\, p_i\, y_i}{P(y = 1 \mid X)}$ $\qquad \because$ if $y_i = 0 \Rightarrow$ numerator $= 0$

$\Rightarrow \quad \boxed{\dfrac{x_i\, p_i\, y_i}{1 - \prod_j (1 - p_j)^{x_j}}}$ $\qquad$ from (a)

c) From the M-step of EM algo :-

$$P(Z_i \mid X_i) \leftarrow \frac{\sum_t^T P(Z_i, X_i \mid X, Y)}{\sum_t^T P(X_i \mid X, Y)}$$

$$P(Z_i \mid X_i) \leftarrow \frac{\sum_t^T P(Z_i, X_i \mid X, Y)}{\sum_t^T I(X_i, x_i^{(t)})}$$

$$P(Z_i \mid X_i) \leftarrow \frac{\sum_t^T P(Z_i, X_i \mid X, Y)}{T_i}$$

[∵ only non-zero X_i will be added]

$$P_i = P(Z_i = 1 \mid X_i = 1)$$

$$\Rightarrow P(Z_i = 1 \mid X_i = 1) = P_i \leftarrow \frac{1}{T_i} \sum_t^T P(Z_i = 1, X_i = 1 \mid X = x^{(t)}, Y = y^{(t)})$$

Q)

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \log P(Y = y^{(t)} \mid X = \vec{x}^{(t)})$$

$$= \frac{1}{T} \sum_{t=1}^{T} \log \left[\left(1 - \prod_{i=1}^{n} (1-P_i)^{x_i^{(t)}}\right)^{y^{(t)}} \left(\prod_{i=1}^{n} (1-P_i)^{x_i^{(t)}}\right)^{(1-y^{(t)})} \right]$$

Q 6.3) d

In [1]:

```
import numpy as np
```

In [2]:

```
xfile = "noisyOrX.txt"
yfile = "noisyOrY.txt"
```

In [3]:

```
def parseFile(filename):
    data = []
    with open(filename) as f:
        for line in f:
            data.append([int(item) for item in line.strip('\n').strip().split()])
    return np.asarray(data)
```

In [48]:

```
x = parseFile(xfile)
y = parseFile(yfile)
```

In [49]:

```
p = np.ones(23) * 0.05
```

In [22]:

```
def P_Y_X(x, p):
    return 1 - np.prod(np.power(1 - p, x), axis=1)
```

In [7]:

```
def E_step(x, y, p):
    num = (x * p) * y
    den = P_Y_X(x, p)
    return num/np.expand_dims(den, axis=1)
```

In [24]:

```
def M_step(post, x):
    p = np.sum(post, axis=0)
    den = np.sum(x, axis=0)
    return np.squeeze(np.expand_dims(p, axis=1)/np.expand_dims(den, axis=1))
```

In [25]:

```
def log_likelihood(x, y, p):
    temp = np.zeros(y.shape)
    for i in range(y.shape[0]):
        temp[i] = P_Y_X(np.expand_dims(x[i], axis=0), p)
        if y[i] == 0:
            temp[i] = 1 - temp[i]
        temp[i] = np.log(temp[i])
    return np.sum(temp, axis=0)/y.shape[0]
```

In [41]:

```
def countMistakes(y, y_pred):
    count = 0
    for i in range(y.shape[0]):
        if y[i] != y_pred[i]:
            count += 1;
    return count;
```

In [46]:

```
peek_iterations = [0, 1, 2, 4, 8, 16, 32, 64, 128, 256]
```

In [50]:

```
for i in range(257):
    y_pred = P_Y_X(x, p)
    y_pred[y_pred >= 0.5] = 1
    y_pred[y_pred < 0.5] = 0
    if i in peek_iterations:
        print(i, countMistakes(y, y_pred), log_likelihood(x, y, p))
    p = M_step(E_step(x, y, p), x)
```

```
0 175 [-0.95808541]
1 56 [-0.49591639]
2 43 [-0.40822082]
4 42 [-0.36461498]
8 44 [-0.34750062]
16 40 [-0.33461705]
32 37 [-0.3225814]
64 37 [-0.3148267]
128 36 [-0.31115585]
256 36 [-0.31016135]
```

6.4

a) $f(x) = \log \cosh(x)$

$$\cosh(x) = \frac{e^{x}+e^{-x}}{2}$$

$$\Rightarrow f'(x) = \frac{1}{\cosh(x)} \times \frac{d}{dx}\left(\frac{e^{x}+e^{-x}}{2}\right)$$

$$= \frac{\sinh(x)}{\cosh(x)} = \tanh(x)$$

$\Rightarrow$ To find the min. :-

$$f'(x) = 0 \Rightarrow \tanh(x) = 0$$

$$\Rightarrow \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = 0$$

$$\Rightarrow e^{2x} = 1$$

$$\Rightarrow x = 0$$

$$f''(x) = \frac{(e^{x}+e^{-x})(e^{x}+e^{-x})-(e^{x}-e^{-x})(e^{x}-e^{-x})}{(e^{x}+e^{-x})^2}$$

$$= \frac{4}{(e^{x}+e^{-x})^2}$$

$$= \frac{1}{\cosh^2(x)} = \operatorname{sech}^2(x)$$

$$f''(0) > 0$$

$\Rightarrow$ $f(0)$ gives minima at $x=0$

b) $$f'(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \tanh(x)$$

$$\Rightarrow f''(x) = \operatorname{sech}^2(x)$$

$$f'''(x) = 4\left(-2\frac{(e^x+e^{-x})(e^x-e^{-x})}{(e^x+e^{-x})^4}\right)$$

$$= -\frac{8(e^x-e^{-x})}{(e^x+e^{-x})^3}$$

To find ~~for~~ maxima of $f''(x)$

$$\Rightarrow f'''(x) = 0$$

$$\Rightarrow -\frac{8(e^x-e^{-x})}{(e^x+e^{-x})^3} = 0$$

$$\Rightarrow x = 0$$

$$f''''(x) = \frac{d}{dx}\left(\frac{-2\sinh(x)}{\cosh^3(x)}\right).$$

$$= -2\left(\frac{\cosh^3(x)\cosh(x) - \sinh(x) \times 3 \times \cosh^2(x)\sinh(x)}{\cosh^6(x)}\right)$$

$$= -2\left(\frac{\cosh^2(x)\left(\cosh^2(x) - 3\sinh^2(x)\right)}{\cosh^6(x)}\right)$$

$$= -2\left(\frac{\cosh^2(x) - 3\sinh^2(x)}{\cosh^6(x)}\right)$$

$$= -2\left(\frac{1 - 2\sinh^2(x)}{\cosh^6(x)}\right)$$

$$f''''(x) = -2\left(\frac{2.5 - \cosh^2(x)}{\cosh^2(x)}\right)$$

$$2.5 - \cosh^2(0) > 0$$

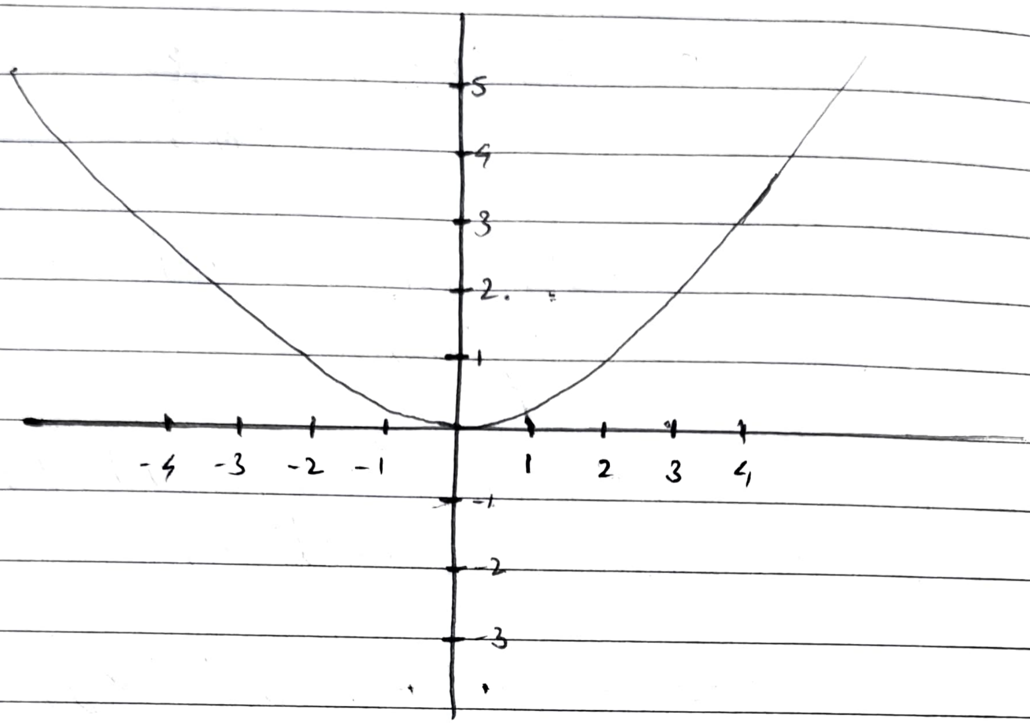
$$\Rightarrow f''''(0) < 0$$

$\Rightarrow f''(0)$ is the ~~fo~~ maxima of $f''(x)$

$$\Rightarrow f''(x) \leq f''(0)$$

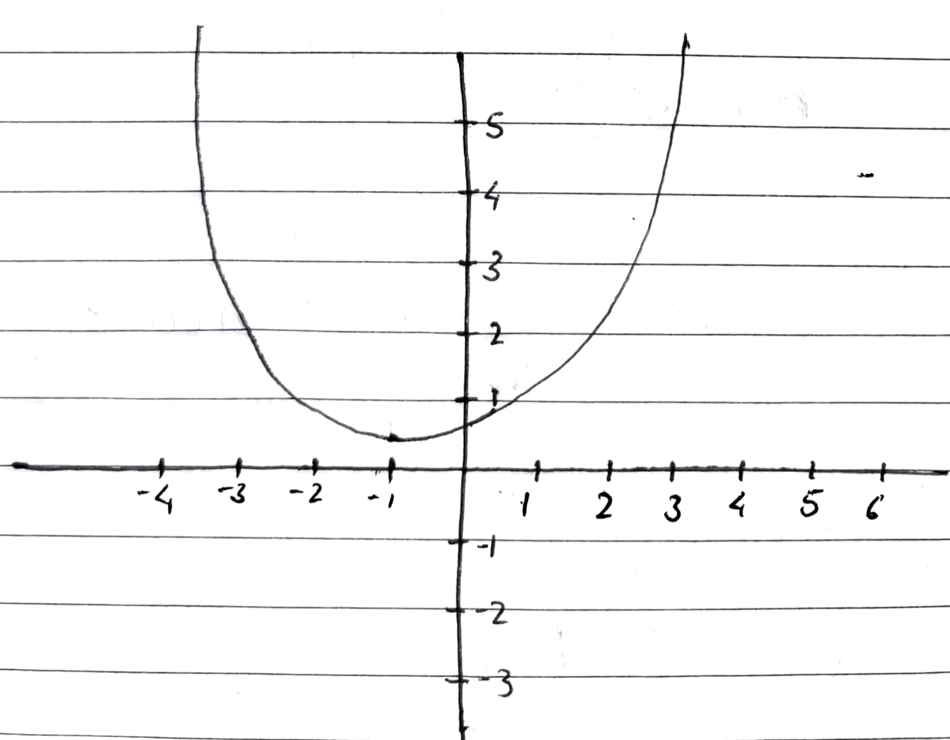
$$\Rightarrow \boxed{f''(x) \leq 1}$$

c) $f(x)$

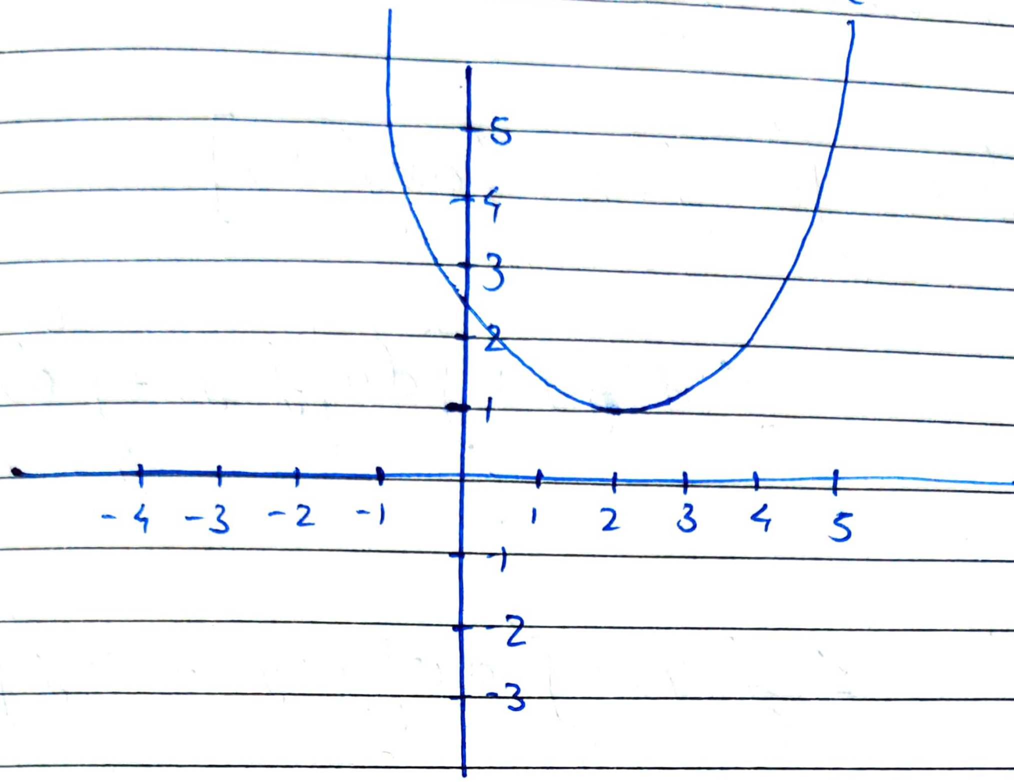


$$\phi(x,y) = f(y) + f'(y)(x-y) + \frac{1}{2}(x-y)^2$$

$$\phi(x,-2) = f(-2) + f'(-2)(x+2) + \frac{1}{2}(x+2)^2$$



$$g(x,3) = f(3) + f'(3)(x-3) + \frac{1}{2}(x-3)^2$$



d)

i) $$g(x,x) = f(x) + f'(x)(x-x) + \frac{1}{2}(x-x)^2$$

$$\boxed{g(x,x) = f(x)}$$

ii) $$f(x) = f(y) + \int_y^x dv\left[f'(y) + \int_y^v dv\, f''(v)\right] \quad \ldots (i)$$

$$\because f''(x) \leq 1$$

$$\Rightarrow \int_y^v dv\, f''(v) \leq \int_y^v dv \quad \ldots (ii)$$

$\Rightarrow$ Substituting (ii) in (i)

$$f(y) + \int_y^x du \left[f'(y) + \int_y^u dv\, f''(v) \right]$$

$$\leq f(y) + \int_y^x du \left[f'(y) + \int_y^u dv \right]$$

$$\Rightarrow f(x) \leq f(y) + \int_y^x du \left[f'(y) + u - y \right]$$

$$\Rightarrow f(x) \leq f(y) + \int_y^x du\, (f'(y)) + \int_y^x (u-y)\, du$$

$$\Rightarrow f(x) \leq f(y) + f'(y)(x-y) + \frac{(x-y)^2}{2}$$

$$\Rightarrow \boxed{f(x) \leq Q(x,y)}$$

Thus, $Q(x,y)$ is an auxiliary function for $f(x)$

e) $x_{n+1} = \underset{x}{\operatorname{argmin}}\ \phi(x, x_n)$

$$= \underset{x}{\operatorname{argmin}}\ f(x_n) + f'(x_n)(x-x_n) + \frac{(x-x_n)^2}{2} \quad \cdots (i)$$

let $g(x) = f(x_n) + f'(x_n)(x-x_n) + \frac{(x-x_n)^2}{2}$

$\Rightarrow g'(x) = f'(x_n) + (x - x_n)$

To find minima:-

$g'(x) = 0$

$\Rightarrow f'(x_n) + (x - x_n) = 0$

$\Rightarrow \boxed{x = x_n - f'(x_n)}$

$\Rightarrow \boxed{x = x_n - \tanh(x_n)}$

$\Rightarrow \boxed{x_{n+1} = x_n - \tanh(x_n)}$ substituting in (i)

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import math
```

In [53]:

```
def fx(x):
    return np.log(np.cosh(x))
    """try:

    except OverflowError as oe:
        return float(math.inf)"""
```

In [54]:

```
def d_fx(x):
    return np.tanh(x)
    """try:

    except OverflowError as oe:
        return float(math.inf)"""
```

In [55]:

```
def d_d_fx(x):
    return 4/pow(2 * np.cosh(x), 2)
    """try:

    except OverflowError as oe:
        return float(math.inf)"""
```

In [56]:

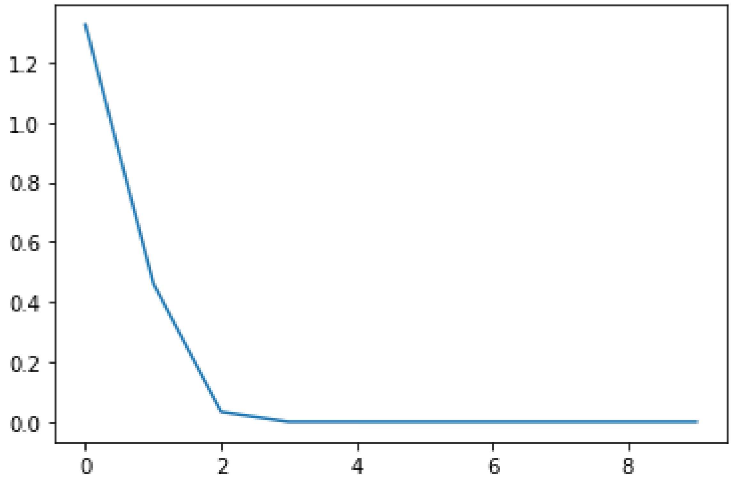
```
def update1(x_0, n):
    f_vals = []
    x = x_0
    for i in range(n):
        f_vals.append(fx(x))
        x = x - d_fx(x)
    return f_vals
```

6.4) f

Typesetting math: 0%

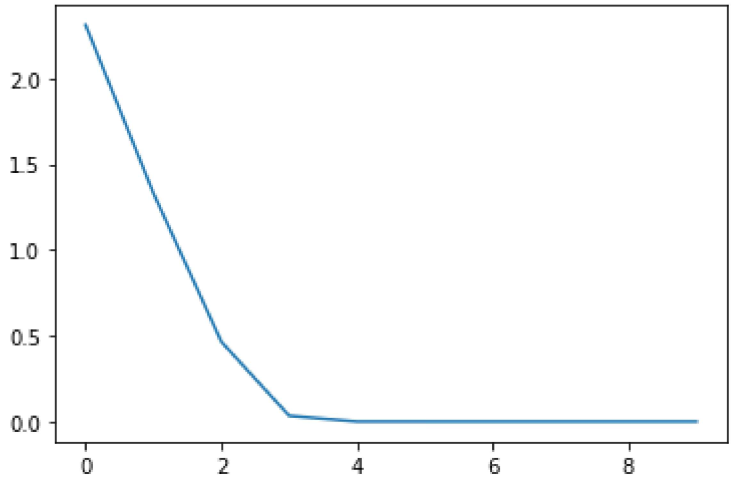
In [57]:

```
plt.plot(update1(-2, 10))
plt.show()
```



In [58]:

```
plt.plot(update1(3, 10))
plt.show()
```



6.4) g

Typesetting math: 0%

(g) $$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}$$

$$\Rightarrow x_{n+1} = x_n - \frac{\tanh(x)}{\operatorname{sech}^2(x)}$$

$$x_{n+1} = x_n - \sinh(x)\cosh(x)$$

$$\Rightarrow x_1 = x_0 - \sinh(x)\cosh(x)$$

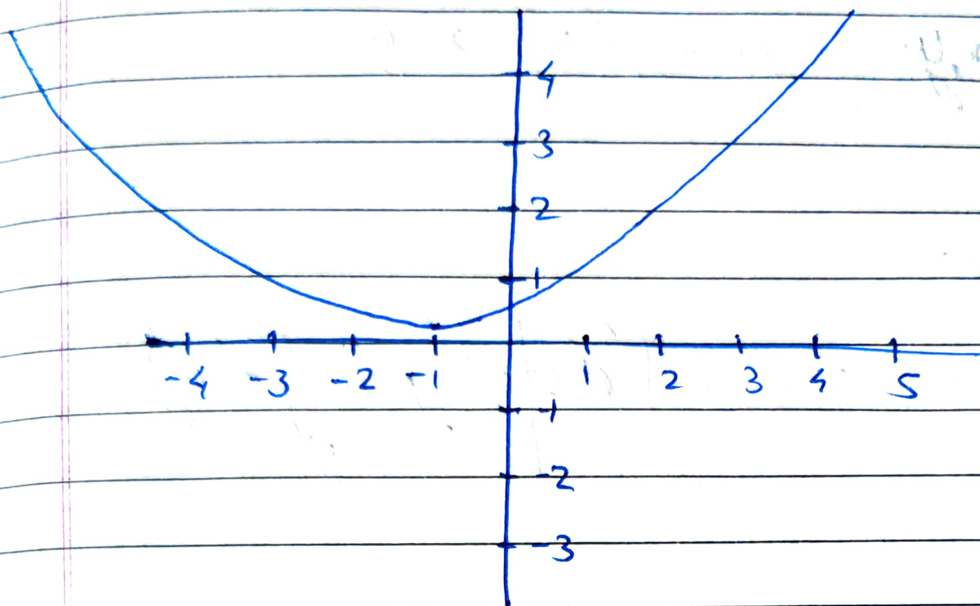
$\Rightarrow$ for convergence

$$|x_1| < |x_0|$$

$$\Rightarrow |x_0 - \sinh(x)\cosh(x)| < |x_0|$$

$$\Rightarrow -1.08866 < x < 1.08866$$

h) $g(x) = \frac{1}{10} \sum_{k=1}^{10} \log\cosh\left(x + \frac{2}{\sqrt{k}}\right)$



No, can't find minima through analytical methods
Thus, will use numerical methods.

(i) $R(x,y) = g(y) + g'(y)(x-y) + \frac{1}{2}(x-y)^2$

1) $R(x,x) = g(x) + g'(x)(x-x) + \frac{1}{2}(x-x)^2$

$R(x,x) = g(x)$

2) $R(x,y) = g(y) + g'(y)(x-y) + \frac{1}{2}(x-y)^2$

$\because f''(x) \le 1 \Rightarrow f''\left(x + \frac{2}{\sqrt{k}}\right) \le 1$

$\Rightarrow \frac{1}{10} \sum_{k=1}^{10} f''\left(x + \frac{2}{\sqrt{k}}\right) \le 1$

$\Rightarrow g''(x) \le 1$

$\Rightarrow$ Similar to part(c)

$$R(x,y) = g(y) + g'(y)(x-y) + \frac{1}{2}(x-y)^2 \quad \text{(i)}$$

Also,

$$g(x) = g(y) + \int_y^x du \left[g'(y) + \int_y^u dv\, g''(v) \right] \quad \text{(ii)}$$

$\because \quad g''(x) \le 1$

$$\Rightarrow \int_y^u dv\, g''(v) \le \int_y^u dv$$

$$\Rightarrow g(x) \le g(y) + \int_y^x du \left[g'(y) + \int_y^u dv \right]$$

$$\Rightarrow g(x) \le g(y) + \int_y^x du \,(g'(y)) + \int_y^x du\,(u-y)$$

$$\Rightarrow g(x) \le g(y) + g'(y)(x-y) + \frac{(x-y)^2}{2}$$

$$\Rightarrow g(x) \le R(x,y)$$

$\Rightarrow$ $R(x,y)$ is auxiliary function to $g(x)$

(i) $x_{n+1} = \underset{x}{\operatorname{argmin}} R(x, x_n)$

$$= \underset{x}{\operatorname{argmin}}\ g(x_n) + g'(x_n)(x-x_n) + \frac{(x-x_n)^2}{2}$$

let,

$$l(x) = g(x_n) + g'(x_n)(x-x_n) + \frac{(x-x_n)^2}{2}$$

$$l'(x) = g'(x_n) + x - x_n$$

To find minima

$l'(x) = 0$

$\Rightarrow$ $0 = g'(x_n) + x - x_n$

$\Rightarrow$ $x = x_n - g'(x)$

$\Rightarrow$ $\boxed{x_{n+1} = x_n - g'(x)}$

In [64]:

```
def update2(x_0, n):
    f_vals = []
    x = x_0
    for i in range(n):
        print(x, fx(x))
        f_vals.append(fx(x))
        x = x - (d_fx(x)/d_d_fx(x))
    return f_vals
```

In [65]:

```
update2(-2, 10)
```

```
-2 1.3250027473578645
11.644958598563875 10.951811418080721
-3255536207.1877036 inf
inf inf
nan nan
nan nan
nan nan
nan nan
nan nan
nan nan

C:\Users\vaibh\AppData\Local\Temp/ipykernel_6836/3182926395.py:2: RuntimeWar
ning: overflow encountered in cosh
  return np.log(np.cosh(x))
C:\Users\vaibh\AppData\Local\Temp/ipykernel_6836/2058034218.py:7: RuntimeWar
ning: divide by zero encountered in double_scalars
  x = x - (d_fx(x)/d_d_fx(x))
C:\Users\vaibh\AppData\Local\Temp/ipykernel_6836/2058034218.py:7: RuntimeWar
ning: invalid value encountered in double_scalars
  x = x - (d_fx(x)/d_d_fx(x))
```

Out[65]:

```
[1.3250027473578645,
 10.951811418080721,
 inf,
 inf,
 nan,
 nan,
 nan,
 nan,
 nan,
 nan]
```

Typesetting math: 0%

In [66]:

```
update2(3, 10)
```

```
3 2.309328504577785
-97.85657868513961 97.16343150457966
2.4836150932578143e+84 inf
-inf inf
nan nan
nan nan
nan nan
nan nan
nan nan
nan nan

C:\Users\vaibh\AppData\Local\Temp/ipykernel_6836/3182926395.py:2: RuntimeWar
ning: overflow encountered in cosh
  return np.log(np.cosh(x))
C:\Users\vaibh\AppData\Local\Temp/ipykernel_6836/2058034218.py:7: RuntimeWar
ning: divide by zero encountered in double_scalars
  x = x - (d_fx(x)/d_d_fx(x))
C:\Users\vaibh\AppData\Local\Temp/ipykernel_6836/2058034218.py:7: RuntimeWar
ning: invalid value encountered in double_scalars
  x = x - (d_fx(x)/d_d_fx(x))
```

Out[66]:

```
[2.309328504577785, 97.16343150457966, inf, inf, nan, nan, nan, nan, nan, na
n]
```

6.4) k

In [67]:

```
def gx(x):
    result = 0
    for k in range(1, 11):
        result += fx(x + (2 / np.sqrt(k)))

    return result/10
```

In [68]:

```
def d_gx(x):
    result = 0
    for k in range(1, 11):
        result += np.tanh(x + (2 / np.sqrt(k)))

    return result/10
```

Typesetting math: 0%

In [69]:

```
def update3(x_0, n):
    g_vals = []
    x = x_0
    for i in range(n):
        g_vals.append(gx(x))
        x = x - d_gx(x)
    return g_vals
```

In [73]:

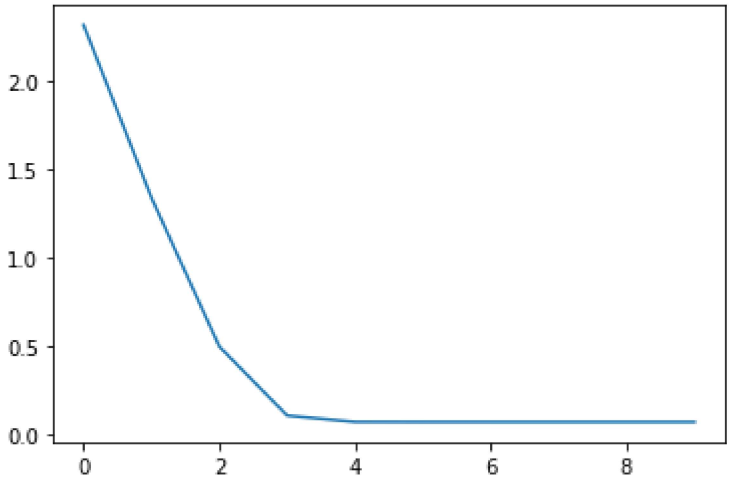
```
update3(2, 10)
```

Out[73]:

```
[2.3141309089320035,
 1.339399484892293,
 0.5005916001821507,
 0.11050257838877894,
 0.07474292148721459,
 0.07421617940304373,
 0.07420940457263744,
 0.07420931571860692,
 0.0742093145501224,
 0.07420931453475131]
```

In [74]:

```
plt.plot(update3(2, 10))
plt.show()
```



In [ ]:

Typesetting math: 0%