Problem 1:

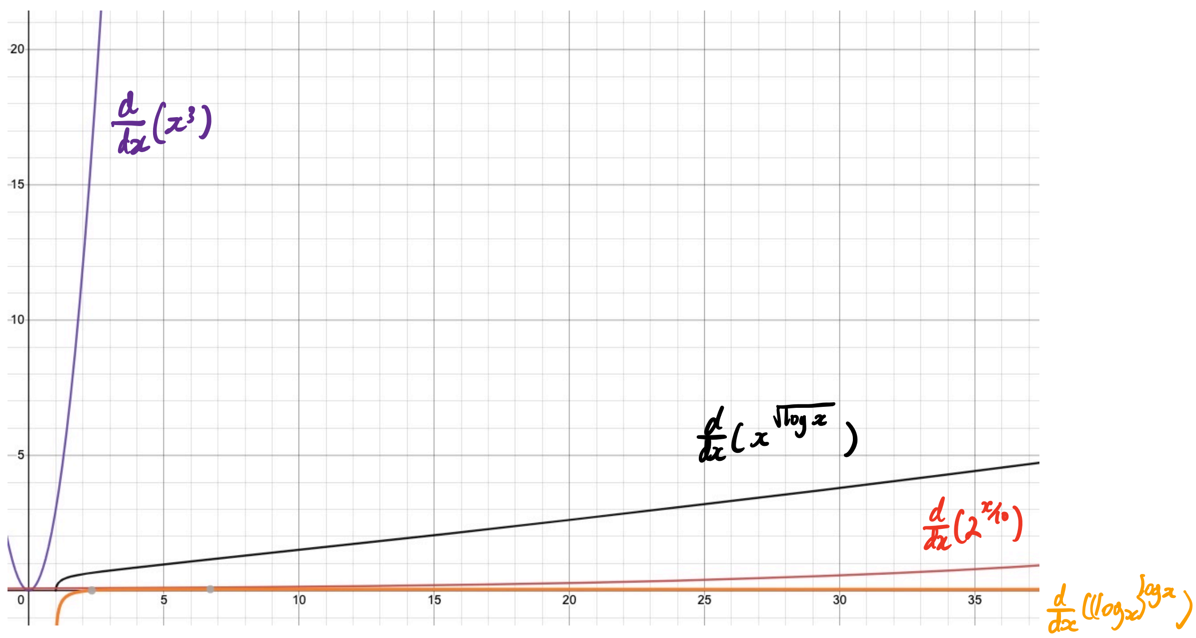
Order : $(\log n)^{\log n} < 2^{\frac{n}{10}} < n^{\sqrt{\log n}} < n^3$

$$n^3 \rightarrow \frac{d}{dn} = 3n^2$$

$$(\log n)^{\log n} \rightarrow \frac{d}{dn}(\log n)^{\log n} = \frac{2e^{\log^2 n}\log n}{n^{\log n + 1}}$$

$$n^{\sqrt{\log n}} \rightarrow \frac{d}{dn} n^{\sqrt{\log n}} = \frac{3n^{\sqrt{\log(n)}-1}\sqrt{\log(n)}}{2}$$

$$2^{\frac{n}{10}} \rightarrow \frac{d}{dn} 2^{\frac{n}{10}} = \frac{1}{10}\log(2)e^{\frac{\log(2)x}{10}}$$



→ if we compare the first order derivatives of each function we can see that n^3 yields the highest growing rate and $(\log n)^{\log n}$ is the lowest. Therefore by comparing their derivatives we can see that the above order is representative of each functions growth rates from smallest to highest

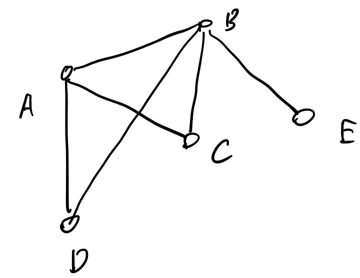
Problem 2:

→ If every vertex in an undirected graph has a degree of $n/2$ this means that every vertex is connected to at least $n/2$ other vertices. This implies that for any 2 vertices in the graph, there must be a path of length 2 or less between them, since each vertex can be reached from any other vertex through at most 2 edges. Therefore the diameter of the graph is at most 2.

→ Every vertex in the graph is connected to half of the graph's total vertexes. So if a graph has nodes (A, B, C, D, E, F) if we look at random nodes A and B for instance:

A → connects to 3 arbitrary nodes $\{B, C, D\}$

B → connects to 3 arbitrary nodes $\{C, D, E\}$



→ if the 2 nodes are also connected we know that any node connected to A has a distance of 1 and any node connected to B has a distance of 1. Therefore the distance between any set of nodes $\{A, B, C, D, E\}$ would have a max distance of $1+1=2$. This also applies to any 2 random nodes we select in graph G

Problem 3 :

Algorithm:

1. Initalize an empty list L, to store the colors of the edges.
2. For each vertex v in V do the following:
   a.) Initalize an empty list, C, to store the colors of the edges incident to v
   b.) For each edge e incident to v, do the following:
      i) assign the edge e color i, where i is the smallest color not already in C.
      ii) append the color i to C
   c.) Append the list C to L

```
L = []
for v in V:
    C = []
    for e in edges incident to v:
        i= smallest color not in C
        assign color i to e
        append i to C
    append C to L
```

→ The algorithm garantees that no 2 edges incident to the same vertex are assigned the same color.

Problem 4:

```python
import random

def coupon_collector(n):
    final = set([i for i in range(1, n + 1)])
    collected = set()
    C = 0
    while collected != final:
        draw = random.randint(1, n)
        collected.add(draw)
        C += 1
    return C, C / n

if __name__ == '__main__':
    f = open('q4.txt', 'w')
    for n in range(200, 4001, 200):
        C, C_avg = coupon_collector(n)
        f.write(f'{n}, {C}, {C_avg}\n')
    f.close()
```

```
n, C, C_avg
200, 1283, 6.415
400, 2290, 5.725
600, 3539, 5.898333333333333
800, 5206, 6.5075
1000, 10025, 10.025
1200, 10350, 8.625
1400, 10917, 7.797857142857143
1600, 9630, 6.01875
1800, 13427, 7.459444444444444
2000, 16054, 8.027
2200, 20421, 9.282272727272728
2400, 21625, 9.010416666666666
2600, 18846, 7.248461538461538
2800, 18454, 6.590714285714285
3000, 23898, 7.966
3200, 27443, 8.5759375
3400, 32755, 9.633823529411766
3600, 27194, 7.553888888888889
3800, 32722, 8.611052631578948
4000, 38714, 9.6785
```

Problem 5:

```python
import random

def coupon_collector(n):
    coupons = {i : float('inf') for i in range(1, n + 1)}
    final = set([i for i in range(1, n + 1)])
    collected = set()
    V = 0
    while collected != final:
        draw = random.randint(1, n)
        value = random.randint(1, n)
        coupons[draw] = min(coupons[draw], value)
        collected.add(draw)
    for v_i in coupons.values():
        V += v_i
    return V, V / n

if __name__ == '__main__':
    f = open('q5.txt', 'w')
    for n in range(200, 4001, 200):
        V, V_avg = coupon_collector(n)
        f.write(f'{n}, {V}, {V_avg}\n')
    f.close()
```

```
n, V, V_avg
200, 8701, 43.505
400, 30048, 75.12
600, 51337, 85.56166666666667
800, 111790, 139.7375
1000, 145204, 145.204
1200, 172464, 143.72
1400, 178512, 127.50857142857143
1600, 370325, 231.453125
1800, 495511, 275.2838888888889
2000, 415294, 207.647
2200, 574384, 261.08363636363634
2400, 788880, 328.7
2600, 770700, 296.4230769230769
2800, 834431, 298.0110714285714
3000, 979312, 326.43733333333336
3200, 1182125, 369.4140625
3400, 1534073, 451.1979411764706
3600, 1591520, 442.0888888888889
3800, 1697000, 446.57894736842104
4000, 1258358, 314.5895
```