

Final Preparation I

1. Estimate the running time of the following algorithms:

```
def algo1(array):
    for i in range(len(array)-1):
        for j in range(i):
            if array[i] > array[i+1]:
                array[i], array[i+1] = array[i+1], array[i]
```

```
def algo2(m, n):
    if n<0 or m<0:
        return m
    while True:
        p = m%n
        algo2(n,p)
```

```
def algo3(x):
    y = 1
    for _ in range(10):
        y = 0.5*(y + x/y)
    return y
```

```
def algo4(array):
    a, b = array[0], array[1]
    for i in range(1, len(array)//2+1):
        if array[2*i]>array[2*i+1]:
            if a < array[2*i+1]:
                a = array[2*i+1]
            if b > array[2*i]:
                b = array[2*i]
        else:
            if a < array[2*i]:
                a = array[2*i]
            if b > array[2*i+1]:
                b = array[2*i+1]
```

2. Solve the 0-1 knapsack problem with the following set of parameters. The maximum capacity is 50.

Item	a	b	c	d	e	f	g	h	i	j	k
Weight	12	12	12	11	10	10	9	9	7	6	5
Value	16	15	14	13	12	11	10	9	8	7	5

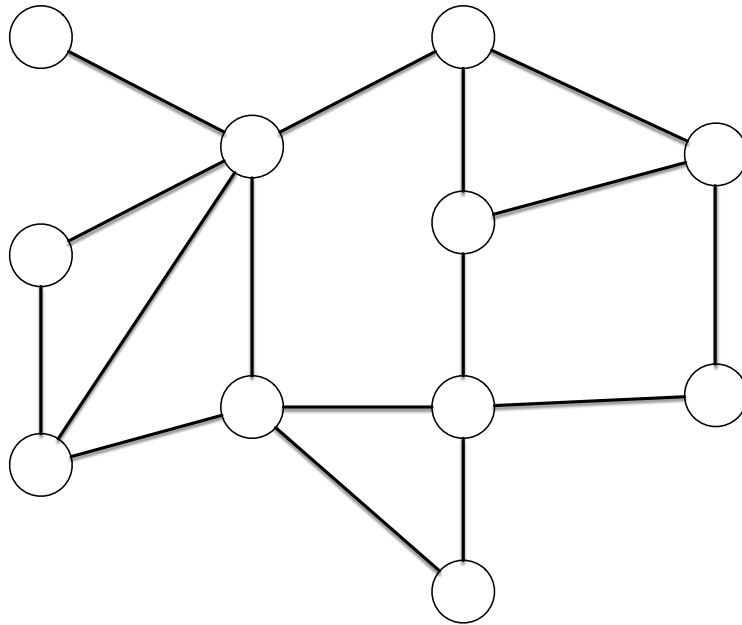
3. Find the least effort way to multiply matrices with the following dimensions

Matrix	Columns	Rows
A	5	10
B	10	12
C	12	10
D	10	2
E	2	20
F	20	5
G	5	5

4. Find the optimal Huffman coding for the following set of symbols and their frequencies:

Symbol	Frequency
a	0.104
b	0.090
c	0.099
d	0.042
e	0.100
f	0.101
g	0.091
h	0.013
i	0.024
j	0.093
k	0.002
l	0.001
m	0.047
n	0.094
o	0.039
p	0.060

5. Use BFS on the following undirected graph. Classify the edges into tree-edges and cross-edges.



6. Show that a BFS search on an undirected graph cannot generate a forward edge.

7. Let $X, Y \subset V$ be two strongly connected subsets of the set of vertices of a directed graph $G = (V, E)$. Assume that there exists an edge with origin in X and destination in Y and another edge with origin in Y and destination in X . Show that $X \cup Y$ is strongly connected.