In [ ]:
```
JOB Scheduling
```

In [1]:
```
def JobScheduling(self, Jobs, n):
    arr = []
    m = -1
    for i in Jobs:
        m = max(m, i.deadline)
        arr.append([i.profit, i.deadline])
    arr.sort(reverse=True)

    job = [-1] * (m + 1)
    for i in arr:
        ind = i[1]
        while ind > 0 and job[ind] != -1:
            ind -= 1
        if ind >= 0:
            job[ind] = i[0]
    job = job[1:]
    x = job.count(-1)
    return [m - x, sum(job) + x]
```

In [ ]:
```
Cycle detection
```

In [ ]:
```
class Graph:
    def __init__(self):
        self.graph = {}

    def add_edge(self, u, v):
        self.graph.setdefault(u, []).append(v)
        self.graph.setdefault(v, []).append(u)

    def print_adjacency_list(self):
        for i in self.graph:
            print(f"{i}: {self.graph[i]}")
```