

### 3. Job Sequencing with Deadlines

Code:

class Job:

```
def __init__(self, job_id, deadline, profit):
```

```
    self.job_id = job_id
```

```
    self.deadline = deadline
```

```
    self.profit = profit
```

```
def job_sequencing(jobs):
```

```
    jobs.sort(key=lambda x: x.profit, reverse=True)
```

```
    n = len(jobs)
```

```
    result = [False] * n
```

```
    job_sequence = [-1] * n
```

```
    for job in jobs:
```

```
        for j in range(min(n, job.deadline) - 1, -1, -1):
```

```
            if not result[j]:
```

```
                result[j] = True
```

```
                job_sequence[j] = job.job_id
```

```
                break
```

```
    return [job_id for job_id in job_sequence if job_id != -1]
```

```
jobs = [Job('a', 2, 100), Job('b', 1, 19), Job('c', 2, 27), Job('d', 1, 25), Job('e', 3, 15)]
```

```
print(job_sequencing(jobs))
```

output:

time complexity:

```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Documents/OriginLab/daa.py
['c', 'a', 'e']
PS C:\Users\karth>
```

Time complexity:

$F(n) = O(n \log n + n * w)$