

# Lecture 04 - Interval Scheduling

February 10, 2021

```
[3]: import random
```

```
[4]: def random_request():  
    return sorted(random.sample(range(100),2))  
  
def make_requests(n):  
    #requests = []  
    #for i in range(n):  
    #    requests.append(random_request())  
    #return requests  
    return [random_request() for i in range(n)]
```

```
[5]: R = make_requests(5)  
print(R)  
print(sorted(R, key=lambda x : x[1]))
```

```
[[24, 51], [26, 52], [16, 45], [29, 70], [27, 62]]  
[[16, 45], [24, 51], [26, 52], [27, 62], [29, 70]]
```

```
[6]: def greedy_solution(requests):  
    sorted_requests = sorted(requests, key=lambda x : x[1])  
    solution = []  
  
    # pop gives you back the element at that index and  
    # removes it from the lists  
    solution.append(sorted_requests.pop(0))  
  
    while len(sorted_requests) > 0:  
        request = sorted_requests.pop(0)  
        # if request can be added to solution, then add it  
        if request[0] >= solution[-1][1]:  
            solution.append(request)  
  
    return solution
```

```
[7]: print(R)  
greedy_solution(R)
```

```
[[24, 51], [26, 52], [16, 45], [29, 70], [27, 62]]
```

```
[7]: [[16, 45]]
```

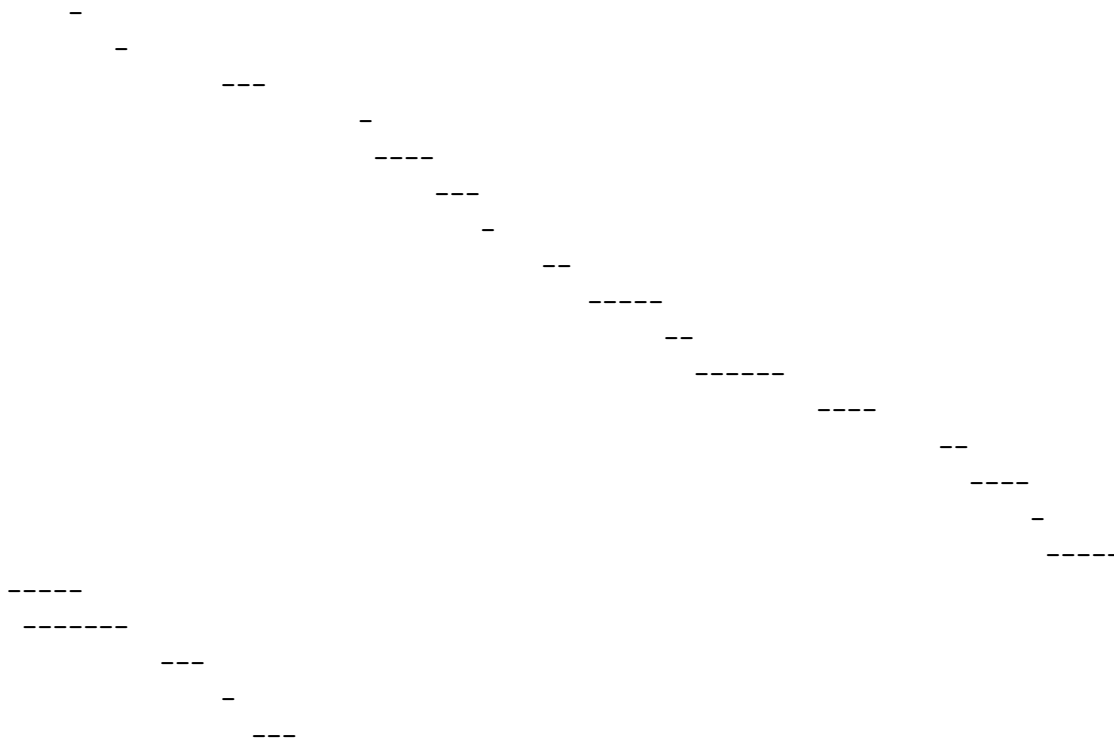
```
[8]: def plot_requests(requests):  
      for r in sorted(requests, key=lambda x:x[1]):  
          print(" " * r[0] + "-" * (r[1]-r[0]))
```

```
[ ]:
```

```
[9]: R = make_requests(500)
```

```
[10]: #plot_requests(R)
```

```
[11]: plot_requests(greedy_solution(R))
```



```
[ ]:
```