

Lecture 04 - Weighted Interval Scheduling Demo

February 17, 2021

```
[1]: import random

def random_request():
    return [sorted(random.sample(range(100),2)), random.random()*10]

def make_requests(n):
    return [random_request() for i in range(n)]

def compatible(r1, r2):
    return r2[0][1] <= r1[0][0] or r2[0][0] >= r1[0][1]

def is_compatible(request, solution):
    r = request
    return all(compatible(r, s) for s in solution)

def greedy(requests, sort_key):
    sorted_requests = sorted(requests, key=sort_key) # O(n*log(n))
    solution = []
    solution.append(sorted_requests.pop(0))

    while len(sorted_requests) > 0: # O(n)
        request = sorted_requests.pop(0)
        if is_compatible(request, solution):
            solution.append(request)

    return solution

def plot_requests(requests):
    # print("="*100)
    for r in sorted(requests, key=lambda x : x[0][1]):
        print(" "*(r[0][0]) + "-"*(r[0][1]-r[0][0]) + " (" + str(round(r[1],2)) +
        ↪")")
```

```
[2]: shortest = lambda x: x[0][1] - x[0][0]
most_value = lambda x: -x[1]
density = lambda x: -(x[1])/(x[0][1] - x[0][0])
```

```
[9]: requests = make_requests(100)
```

```
[10]: s1 = greedy(requests, shortest)
      s2 = greedy(requests, most_value)
      s3 = greedy(requests, density)
```

```
[11]: print(sum(r[1] for r in s1))
      print(sum(r[1] for r in s2))
      print(sum(r[1] for r in s3))
```

```
53.28910208845178
55.724024509610764
59.42741851476154
```

```
[12]: plot_requests(s1)
```

```
      -- (5.15)
         ----- (1.76)
            ----- (1.76)
               - (1.57)
                  - (0.83)
                     ---- (6.92)
                        ----- (6.78)
                           -----
(8.36)
(7.52)
   --- (9.42)
      -- (3.23)
```

```
[13]: plot_requests(s2)
```

```
      -- (5.15)
         ----- (9.96)
            ----- (6.92)
               ----- (3.93)
                  ----- (9.6)
                     -----
(7.52)
   --- (9.42)
      -- (3.23)
```

```
[14]: plot_requests(s3)
```

```
      -- (5.15)
         ----- (9.65)
            - (1.57)
               - (0.83)
                  ---- (6.92)
                     ----- (6.78)
                        -----
```

(8.36)

(7.52)

--- (9.42)

-- (3.23)

[]: