Lecture 03 - Interval Scheduling (in-progress)

February 3, 2023

[1]:

```python
import random

def random_request():
    "generates a single random request"
    return sorted(random.sample(range(0,100), 2))

def make_requests(n):
    "generates n random requests"
    # slower version
    #requests = []
    #for i in range(n):
    #    requests.append(random_request())
    #return requests
    # faster version:
    return [random_request() for i in range(n)]

def greedy_solution(requests):
    """
    input is a set of requests, output is the
    greedy solution where best = earliest end
    time
    """
    sorted_requests = sorted(requests, key=lambda x : x[1])

    solution = []
    solution.append(sorted_requests.pop(0))
    # when you call L.pop(i) for a list L, it removes
    # the element at index i and returns it for you

    # TODO: remove conflicts with this chosen meeting
    # and repeat until there are no sorted_requests
    while len(sorted_requests) > 0:
        # look at the first thing in sorted_requests
        # if it conflicts with a meeting we already accepted,
        #    throw it away
        # otherwise, add it to our solution
        request = sorted_requests.pop(0)
        # is it good?
        if request[0] >= solution[-1][1]:
            # request[0] is the start of the request we're considering
            # solution[-1] is the last request we have in our solution
            # (solution[-1])[1] is its end time
            solution.append(request)

    return solution
```

[3]:
```
requests = make_requests(10)
print(requests)
```

```
[[52, 82], [89, 95], [84, 87], [36, 73], [42, 57], [48, 59], [46, 63], [76, 94],
[4, 53], [37, 41]]
```

[4]:
```
greedy_solution(requests)
```

[4]:
```
[[37, 41], [42, 57], [84, 87], [89, 95]]
```

[11]:
```
def plot_requests(requests):
    for r in sorted(requests, key=lambda x : x[1]):
        print(" "*(r[0]) + "-"*(r[1]-r[0]))
```

[13]:
```
plot_requests([[2,5],[4,7],[1,10]])
```

```
  ---
    ---
 ---------
```

[26]:
```
R = make_requests(100_000)
```

[27]:
```
#plot_requests(R)
```

[28]:
```
A = greedy_solution(R)
print(A)
print(len(A))
```

```
[[0, 1], [1, 2], [2, 3], [3, 4], [4, 5], [5, 6], [6, 7], [7, 8], [8, 9], [9,
10], [10, 11], [11, 12], [12, 13], [13, 14], [14, 15], [15, 16], [16, 17], [17,
18], [18, 19], [19, 20], [20, 21], [21, 22], [22, 23], [23, 24], [24, 25], [25,
26], [26, 27], [27, 28], [28, 29], [29, 30], [30, 31], [31, 32], [32, 33], [33,
34], [34, 35], [35, 36], [36, 37], [37, 38], [38, 39], [39, 40], [40, 41], [41,
42], [42, 43], [43, 44], [44, 45], [45, 46], [46, 47], [47, 48], [48, 49], [49,
50], [50, 51], [51, 52], [52, 53], [53, 54], [54, 55], [55, 56], [56, 57], [57,
58], [58, 59], [59, 60], [60, 61], [61, 62], [62, 63], [63, 64], [64, 65], [65,
66], [66, 67], [67, 68], [68, 69], [69, 70], [70, 71], [71, 72], [72, 73], [73,
74], [74, 75], [75, 76], [76, 77], [77, 78], [78, 79], [79, 80], [80, 81], [81,
82], [82, 83], [83, 84], [84, 85], [85, 86], [86, 87], [87, 88], [88, 89], [89,
90], [90, 91], [91, 92], [92, 93], [93, 94], [94, 95], [95, 96], [96, 97], [97,
98], [98, 99]]
99
```

[29]:

```
plot_requests(A)
```

```
-
 -
  -
   -
    -
     -
      -
       -
        -
         -
          -
           -
            -
             -
              -
               -
                -
                 -
                  -
                   -
                    -
                     -
                      -
                       -
                        -
                         -
                          -
                           -
                            -
                             -
                              -
                               -
                                -
                                 -
                                  -
                                   -
                                    -
                                     -
                                      -
                                       -
                                        -
                                         -
                                          -
```

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

–

```
           _
            _
             _
              _
               _
                _
                 _
                  _
```

```
[ ]:
```