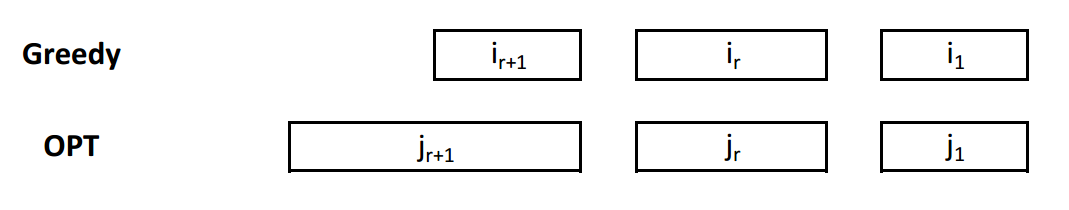
1)

The correctness of this algorithm can be proof by using the “Greedy stays ahead” proof stategy.

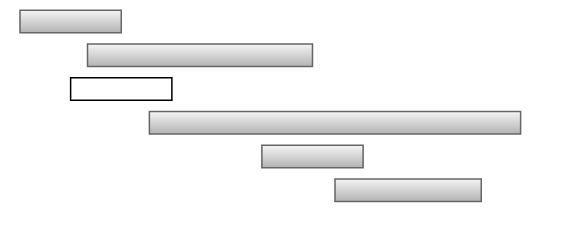
* Suppose that greedy is not optimal. Let:
  + I1, i2 , i3 …. Ik denotes the set of jobs selected by greedy
  + J1, j2 , j3 …….  Jk denotes the set of jobs that is an Optimal solution that agrees with as many initial jobs of greedy solution as possible
  + Let r be the largest index such that
  + Let the figure below depicts the scheduling of the jobs up to job r+1



* + - cannot start before , and cannot finishes before because this would allow you to improve on OPT by adding .
  + Our goal at this point is to come up with another OPT’ that agrees with more initial jobs than r
  + We claim that is another optimal solution.
    - Since starts after , OPT’ does not have more conflicts than OPT
    - **OPT’ agrees on more than r initial jobs with greedy**
  + OPT’ contradicts with OPT, which was supposed to be the optimal solution that agrees on the most initial jobs with greedy.

2.a)

Below is an example where the “most length not covered first” strategy would not work. The grey intervals are the intervals that the algorithm would select.



2.b)



2.c)

The runtime of the algorithm is

2.d)

An important aspect of the algorithm is that the intervals are sorted by decreasing ending time, or B points.

The algorithm is correct because, at each points, the algorithm only takes into consideration only the intervals that are compatible with the intervals that have been selected so far. Selection of which interval from these compatible intervals is based on which of the intervals has the minimum the starting time (or A point). This selection of the minimum starting time allows the algorithm to be correct because an interval with the earliest starting time will always cover the furthest space.