ESERCIZIO 3 di GD

Supponiamo di avere un insieme di attività $A=\{A_1, A_2 \dots A_n\}$ dove ciascuna attività A_i ha un tempo di inizio s_i ed un tempo di fine f_i : $s_i < f_i$. Due attività sono compatibili se $[s_i, f_i] \cap [s_j, f_j]$. In pratica, possono essere eseguite ~~ottimo~~ alle risorse, attività che non si sovrappongono. L'algoritmo calcola ~~ca~~ l'insieme $S \subseteq A$ di attività a due a due compatibili, di cardinalità massima.

L'algoritmo deve quindi creare la soluzione iterativamente, scegliendo ad ogni passo la prima attività che non si interseca con le altre attività, con la condizione che le attività siano ordinate in base ai loro tempi di fine.

```
Greedy-ACTIVITY_SELECTOR(s1 .. sn, f1 .. fn)
 ORDINA LE ATTIVITA' IN ACCORDO a fi
  S <- {A1}, j <- 1
  for i <- 2 to n
     if s_i >= f_j then
       S <- S U {Ai}
       j <- i
  return S
```

Complessità $\Theta(n \log n)$

Per provare la correttezza dell'algoritmo, supponiamo per contraddizione, che la soluzione prodotta non è la migliore (OPT)

Siano $A_{i_1} \dots A_{i_k}$ le attività selezionate da GAS

Siano $A_{j_1} \dots A_{j_m}$ le attività in una soluzione ottima (OPT) con $z-1$ il più grande intero per cui $A_{i_1} = A_{j_1} \dots, A_{z-1} = A_{j_{z-1}}$

Allora A_{i_z} termina prima di A_{j_z}, perché allora non sostituiamo A_{j_z} con A_{i_z}, la soluzione rimane ottima, ma contraddice la massimalità di $z-1$.