Algoritmi greedy
"Un programma affollato di eventi"
Il problema dello scheduling

Introduzione

Questa attività viene proposta dopo quella sul problema del resto ("Monete golose").
Siamo a un festival del cinema o a una manifestazione e vogliamo partecipare al maggior numero di eventi possibli, con il vincolo che se scegliamo un evento, dobbiamo seguirlo dall'inizio alla fine e quindi non possiamo partecipare a eventi che si sovrappongono. Come possiamo scegliere gli eventi a cui partecipare?
Obiettivi generali:

- rendersi conto che problemi molto diversi (il problema del resto e quello dello scheduling) condividono una stessa struttura;
- vedere come si possa stabilire se la strategia greedy è ottima o no.

Scaletta

- **Presentare** il contesto e il problema (vedi sopra).
- Illustrare le **caratteristiche** del problema:
  - gli eventi hanno durate diverse
  - alcuni si sovrappongono
  - ciascuno ha un orario fissato (inizio e fine)
  - ogni evento viene proposto una sola volta
  - l'**obiettivo** è assistere al maggior numero possibile di eventi.
- Far emergere che sono "**ammissibili**" i sottoinsiemi di eventi che non si sovrappongono mai, e che si deve quindi trovare un sottoinsieme ammissibile "con più eventi possibili".
- **Richiamare la procedura**: esamino un evento per volta e decido se usarlo oppure no per comporre il mio programma di eventi cui partecipare (un evento scartato non verrà piú considerato), cioè:
  1. ordino tutti gli oggetti (eventi) in base a un certo CRITERIO;
  2. considero gli oggetti uno alla volta, nell'ordine definito al punto 1 e, per ogni oggetto X considerato:
     - se VALE una certa CONDIZIONE, allora uso X, (qui: l'evento X non si sovrappone agli eventi già scelti)
     - altrimenti lo scarto (un oggetto scartato non verrà più considerato).

  In che ordine esamino gli eventi? Qual è il criterio da usare per ordinare gli eventi in questo problema?
- Far emergere a classe intera i possibili **criteri di ordinamento**. Dovrebbero emergere:
  - durata
  - orario di inizio
  - orario di fine
  - numero di sovrapposizioni con altri eventi
- **Distribuire la scheda** "Un programma affollato di eventi" e presentare la soluzione di Ale Gridi.
- **Illustrare il software** per fare esperimenti:
  - il software visualizza graficamente orario di inizio e di fine di un numero prefissato di eventi. uno per riga;
  - è possibile generare gli orari in modo casuale o intervenire manualmente per cancellare, aggiungere, modificare gli eventi

  - cliccando sul bottone "Scegli il criterio di ordinamento" è possibile scegliere un criterio di ordinamento tra alcune alternative prefissate (quello che inizia prima, il più corto, quello con meno sovrapposizioni, ...)
  - cliccando sul bottone "Determina gli eventi cui partecipare" viene applicala la strategia di Ale Gridi che sceglie gli eventi a cui assistere in base al criterio fissato, vengono visualizzati gli eventi riordinati in base al criterio fissato, uno per riga, e presentato il risultato della procedura greedy: partendo dal primo, gli eventi sono selezionati (colorati in verde) se non si sovrappongono ai precedenti; quelli in rosso sono gli eventi scartati. Viene inoltre mostrato il numero di eventi selezionati.
- Una volta completata la scheda, **far condividere** l'esito delle loro **riflessioni**:
  - quali criteri hanno escluso e perché (come li hanno individuati?)
  - se ci sono arrivati, perché il criterio "finisce prima" è buono
  - altrimenti spiegazione da parte del conduttore: confronto tra soluzione ottima A e soluzione greedy G con il criterio "finisce prima", dove gli eventi delle due soluzioni sono elencati in ordine in base a questo criterio. Osservare che il primo evento di G finisce di sicuro prima del primo di A, il secondo evento di G finisce di sicuro prima del secondo di A (altrimenti avrei scelto il secondo di A), e così via. In particolare questo succede con l'ultimo di G. A non può avere ulteriori intervalli altrimenti sarebbero stati scelti anche in G. Vedi scheda con le soluzioni.

  Nota. Come hanno usato il software? Hanno realizzato che per escludere un criterio, basta un contro-esempio e che invece per accettarlo occorre una dimostrazione? Come hanno costruito i controesempi?