Laboratorio "Informatica per ghiottoni"

Introduzione agli algoritmi greedy

Parte I - Attività sul problema del resto

Scaletta

- **Materiale**: set di tagli di valuta (foglietti stampati, possibilmente su carta di colori diversi per distinguerli più facilmente) buone per la strategia greedy. Ad esempio 1, 2, 5, 10, 20, 50 centesimi (o euro). Software: scheduling o scratch.
- **Presentazione** di Aladdin e dell'attività.
- **Distribuzione post-it**: come definiresti "elaborazione"? (nel contesto di "informatica = elaborazione automatica dell'informazione")
- Raccolta post-it, **clusterizzazione** e lettura
- **Avvicinamento**: chiedere ai ragazzi di dire di che tagli è composto un resto di 82 euro (questo in modo che tutti abbiano la consapevolezza che si tratta di un problema che loro sanno risolvere)
- **Gioco algomotorio**: divisi in gruppi da 3/4 (un gruppo per tavolo) sviluppare un programma per una cassa automatica (es supermercato/macchinette) che deve dare il resto. Regole del gioco: il cassiere deve dare il resto utilizzando meno monete/banconote possibile. Diamo a ogni gruppo le monete/banconote finte per aiutarsi. Scrivere (in italiano, niente post-it) la procedura che deve eseguire la cassa automatica. Inizialmente ipotizzare che la cassa contenga una disponibilità infinita di pezzi per ogni taglio. Se emerge solo la soluzione con divisione e modulo, aggiungere il vincolo che la cassa non sa fare la divisione.
- **Fusione dei gruppi** a due a due con il compito di convergere su una procedura condivisa.
- **Lettura ad alta voce delle soluzioni** e schematizzazione per ragionare sugli algoritmi greedy e far emergere gli aspetti comuni, che dovrebbero essere sostanzialmente questi:
  - Esamino un "pezzo" per volta e decido se usarlo oppure no per comporre il resto.
    In base a quale criterio decido se usarlo o no?
    (un "pezzo" scartato non verrà più considerato).
  - In che ordine esamino i "pezzi"?
    In base al valore, partendo da quella di valore maggiore
  - Se li metto in un altro ordine, ottengo sempre lo stesso risultato usando questa procedura?
    Cosa cambia?
- **Riformulazione** da parte del conduttore del problema e della procedura in maniera più precisa, che fa emergere i concetti di *soluzione ammissibile* e *soluzione ottima*.
  Problema del resto:
  - ho un insieme di pezzi di tagli diversi
  - i sottoinsiemi di pezzi con valore complessivo pari al resto da comporre sono "sottoinsiemi ammissibili"
  - devo trovare il sottoinsieme ammissibile che usa il minor numero di pezzi possibile

  Procedura greedy:
  1. Ordino tutti i pezzi in base al valore, partendo da quello di valore maggiore.
  2. Considero i pezzi uno alla volta, nell'ordine definito al punto 1 e, per ogni pezzo X considerato:
     Se il valore di X non supera la parte di resto che ancora devo comporre,
     allora uso X,
     altrimenti la scarto.

  Nota: posso anche evitare di esaminare tutti i pezzi, fermandomi una volta che ho già raggiunto una valore complessivo pari al resto.

- **Formulazione piú precisa** basata sui tagli, che potrebbe essere emersa dai gruppi:
  1. metto *in ordine* i soldi nella cassa organizzandoli a blocchetti di tagli di ugual valore che dispongo in ordine decrescente di valore
  2. analizzo i tagli uno alla volta, nell'ordine (dal taglio di valore piú alto a quello di valore piú basso (*iterazione*)
     - se il taglio che sto considerando è minore o uguale al resto da dare (*ammissibilità*), lo uso, aggiorno il resto da dare (*aggiornamento della soluzione*) e ripeto questa istruzione
     - altrimenti lo scarto e non lo prendo piú in considerazione (vado al taglio successivo),
  3. mi fermo quando il resto da dare è zero
- **Problema generale di ottimizzazione**:
  - ho un insieme di oggetti/elementi (es: monete di tagli diversi)
  - alcuni sottoinsiemi di questo oggetti sono "ammissibili" (es: gruppi di monete con valore complessivo pari al resto da comporre)
  - devo trovare un sottoinsieme ammissibile "ottimo" (es: che usa il minor numero di monete)
- **Schema generale di soluzione greedy**
  1. Ordino tutte gli oggetti in base a un certo CRITERIO (es: valore decrescente delle monete)
  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, (es: la moneta X è minore o uguale al resto ancora da dare)
     - altrimenti lo scarto (un oggetto scartato non verrà più considerato).

  Il CRITERIO di ordinamento e la CONDIZIONE per cui usare/scartare l'oggetto cambieranno a seconda del problema.
- Introduzione del sistema monetario inglese in vigore fino al 1971 (**esempio** di tagli per cui **non** vale la procedura **greedy**), con monete da
  - 30 pence
  - 24 pence
  - 12 pence
  - 6 pence
  - 1 penny
- Sfida tra squadre, a chi trova la composizione minima in monete degli importi di 17 pence (6 monete: $12 + 1 + 1 + 1 + 1 + 1$), 21 pence (5 monete: $12 + 6 + 1 + 1 + 1$) e 48 pence (2 monete: $12 + 12$, ma l'algoritmo greedy userebbe 3 monete: $30 + 12 + 6$).
- Si chiede a ogni squadra di applicare il loro programma e confrontare l'output con le risposte trovate al punto precedente.

Possibili fasi successive del laboratorio

1. Programmazione in Scratch della macchina del resto.
2. Il problema dello scheduling, seguendo la scheda "Un programma affollato di eventi".