Stable Matching

Algoritmo che si occupa di matchare coppie di elementi in base alle loro esigenze

Prendiamo A e B come due insiemi disgiunti:
$|A| = |B| = n$, $A \cap B = \emptyset$

Vogliamo creare delle coppie $\{a, b\}$ tc $a \in A$ e $b \in B$ rispettando le preferenze

Esempio

$A = \{a_1, a_2, a_3, a_4\}$

$B = \{b_1, b_2, b_3, b_4\}$

$a_1 : b_2 > b_1 \longrightarrow a_1$ preferisce b_2 a b_1

$b_2 : a_1 > a_2 \longrightarrow b_2$ preferisce a_1 a a_2

Possiamo creare la coppia $\{a_1, b_2\}$

Def (Matching perfetto)

È un accoppiamento tra gli elementi di A con gli elementi di B
Sia $M \subseteq \{(a, b) \text{ tc } a \in A, b \in B\}$ M è un matching perfetto $\Leftrightarrow \forall a \in A \; \exists b \in B$ tc $\{a, b\} \in M$, quindi $\forall b \in B \; \exists a \in A$ tc $\{a, b\} \in M$

Def (Matching stabile)

Si suppone che ogni $a \in A$ abbia una lista di preferenza su ogni $b \in B$ e analogamente ogni $b \in B$ ha una propria lista

Sia M un matching perfetto da A a B, due coppie distinte $\{a, b\}, \{a', b'\} \in M$ hanno instabilità se

- a preferisce b' a b
- b' preferisce a ad a'

Un matching è stabile se non esibisce instabilità, esiste sempre un matching stabile

↓

Algoritmo di Gale Shapley

- Inizialmente ogni $a \in A$ e $b \in B$ sono liberi
- while $\exists a_i \in A$ libero che non si è proposto $\forall b_j \in B$
  - Sia $a_i \in A$ un elemento libero di A che non si è proposto a ogni $b \in B$
  - Sia $B' \subseteq B$ l'insieme degli elementi a cui non si è proposto a_i $B' \neq \emptyset$
  - Sia $b_j \in B'$ l'elemento che è più in alto nella lista di preferenze di a_i

    // a_i si propone a b_j

- if b_j è Libero
  - accoppiamo a_i e b_j $\Rightarrow$
    $\Rightarrow$ a_i e b_j non sono più Liberi

- else b_j non è Libero
  - sia a_k partner di b_j
  - if b_j preferisce a_k $\Rightarrow$
    $\Rightarrow$ b_j rifiuta
    a_i rimane Libero
    rimane $\{a_i, b_j\}$
  - else b_j Preferisce a_i $\Rightarrow$
    $\Rightarrow$ b_j accetta
    a_k è Libero $\rightarrow$ si rompe $\{b_j, a_k\}$
    si crea $\{b_j, a_i\}$