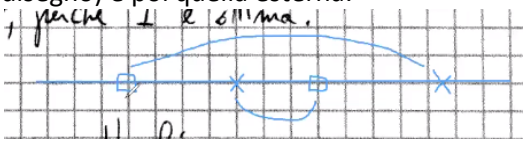
Esercizio → metric matching on the line

Sia $S = s_1, s_2, \dots, s_n$ un insieme di punti ordinati sulla retta reale, rappresentanti dei server.
Sia $C = c_1, c_2, \dots, c_n$ un insieme di punti ordinati sulla retta reale, rappresentanti dei client.
Il costo di assegnare un client c_i ad un server s_j è $|c_i - s_j|$. Si fornisca un algoritmo greedy che assegna ogni client ad un server distinto e che minimizzi il costo totale dell'assegnamento.

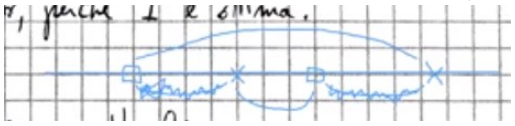
Soluzione

Si vuole minimizzare la somma degli assegnamenti:

- Cerco la coppia client-server più vicina e ripeto (non funziona)
  Vediamo l'esempio di una coppia di client, dove si assegna la coppia più vicina (quella interna nel disegno) e poi quella esterna:

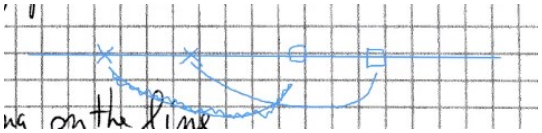


  La soluzione ottima invece farebbe così, cioè assegna client/server a coppie:



Il numero di client e di server è sempre n, quindi ogni client sarà assegnato ad un server.
Cosa si fa: *il primo client a sx viene assegnato al primo server a sx.*
È un algoritmo ottimo, nonostante non consideri le distanze.
Semplicemente, prende il primo client che trova e lo matcha con il primo server che trova.

Anche con la tecnica del *cut&paste*, si ha che la soluzione ottima (quella non tratteggiata) rimane quella buona, in quanto comunque il costo di assegnazione rimane $\leq$ al precedente.



Alla luce di questo, l'algoritmo greedy è il seguente:

$METRIC - MATCHING(S, C)$
$n = C.length$
$A = C[1] - S[1]$ # $assegno\ la\ prima\ coppia\ client - server$
$last = 1$
$for\ i = 2\ to\ n$
    $for\ j = 1\ to\ n - 1$
        $if\ C_i - S_j > C_{(last)} - S_{(last)} + 1$
            $last = i$ # $assegno\ il\ client\ i - esimo\ al\ server\ i - esimo$
            $A = [C_i - S_j] \cup A$
$return\ A$