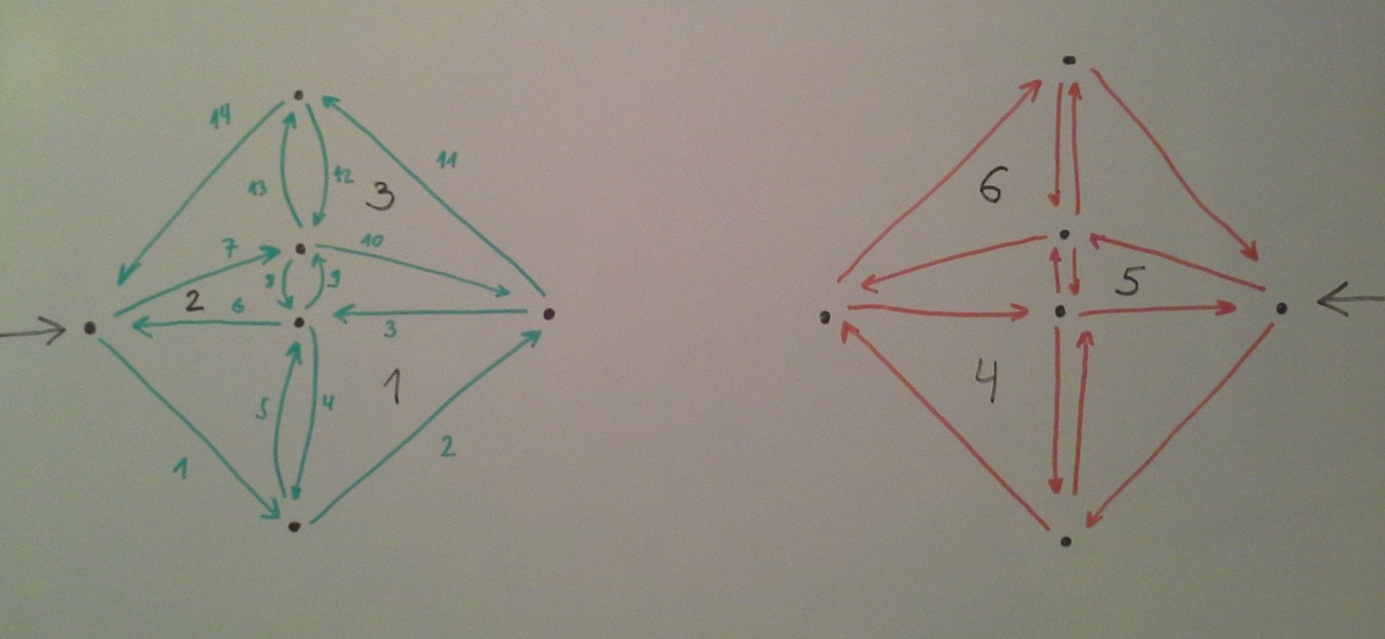
Ovaj algoritam rješava zadani primjer, ali samo kao jako poseban podslučaj općenitih 2D mesheva u ravnini.  
To je slučaj kada imamo dvije točke, a okomito na njihovu ravninu uzmemo fiksno jedan pravac po kojem možemo nizati „beskonačno“ mnogo točaka.  
  
Za općeniti slučaj bi se trebalo, kao prvo, omogućiti unošenje proizvoljnih točaka od strane korisnika. Čim se unesu tri nekolinearne točke koje bi činile prvi trokut – poligon. Svaka iduća točka bi se s ostalim spajala po točno određenim pravilima, ovisno je li dodana unutar poligona, na rub poligona ili izvan. Trebalo bi se paziti da se linije spajanja ne križaju kako ne bismo dobili nove točke sjecišta koje zapravo ne postoje itd... Za općenito rješenje, zadani zadatak postaje iznimno kompleksan.  
Ako pretpostavimo ovaj podslučaj, lakše možemo riješiti zadani primjer.  
  
a) Gledamo svaku od n točaka s ostalih n-1. Pri povezivanju moramo paziti da je spajamo s njoj najbližom točkom ako ih ima više na istom pravcu. I to treba provjeravati u sva četiri smjera. Duplikate izbacujemo i to tako da pripazimo da naš vektor edgeova bude sortiran po prvoj od točaka.  
  
b) Udaljenost dviju točaka u koordinatnom Kartezijevom sustavu

c) Zbog svojstva sortiranosti zadanih točaka po kojem su definirani trokuti i sortiranosti prethodno pronađenih edgeova po prvoj točki, direktno uspoređujemo edgeove trokuta s najmanjim edgeom  
  
Posebno, zadatak se mogao riješiti i bez da su nam zadani trokuti u ovakvom podslučaju.  
To bi se odradilo u dva kraća algoritma.  
1) Krenuli bismo od krajnje lijeve točke, a iduću bi odabirali gdje je y-koordinata najmanja – ako su iste, x-koordinata nove točke morala bi biti veća. Uvijek bi pamtili po 4 točke uzastopno i gledali jesmo li se vratili u prvu od te 4, jer ako jesmo onda se radi o trokutu ! Također bi morali pamtiti da ne ide putem kojim smo već prošli, tada bi se vratili nazad u točku i gledali ima li još puteva iz nje po početnom pravilu.  
  
2) isto kao i pod jedan ali počinjemo od krajnje desne, gledamo minimalnu y-koordinatu, ali uvjet za x-koordinatu je obrnut.



d) Za brisanje trokuta 0,1,2 i 1,2,3 koji dijele edge 1,2:  
  
izbrišemo samo edge 1,2 iz vectora svih edgeova  
  
nađemo točku, nazovimo je A, koja je dio većeg novonastalog trokuta koji je nastao stapljanjem ona dva manja zbog brisanja edgea 1,2, a koja jedina nije kolinearna s ostalima – ili joj je x-koordinata drugačija od tih ostalih točaka ili y-koordinata  
  
kad smo je našli nađemo drugu točku, nazovimo je B, koja je nekad bila dio edgea 1,2 a to možemo po tome što ona ima suprotnu koordinatu iz prethodnog koraka istu kao točka A  
  
zapamtimo sve trenutne veze točke A s drugim točkama  
  
točku A poistovjetimo s B, i A izbrišemo  
  
sve veze s bivšim A preslikamo da su sada veze s B