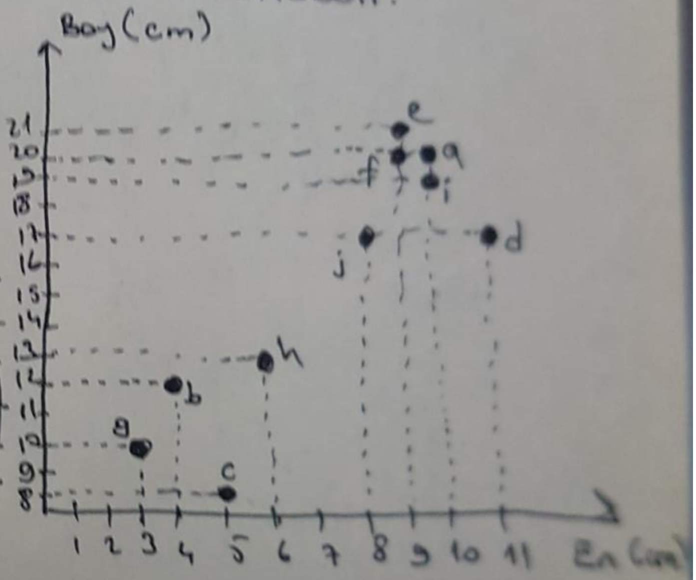
Bir örnek üzerinden kruskal algoritması ile nasıl kümeleme yapıldığını gösterelim.
Bir çiftlikte bulunan balıkların genişlik ve boy bilgilerine göre kümelenmesi istenmektedir. Ölçüm değerleri ve elde edilen değerlerin grafikte gösterimi şu şekildedir.

Balık	En	Boy
1(a)	10	20
2(b)	4	12
3(c)	5	8
4(d)	11	17
5(e)	9	21
6(f)	9	20
7(g)	3	10
8(h)	6	13
9(i)	10	19
10(j)	8	17



Kümeleme işlemi için kruskal algoritması kullanılması için öncelikle bütün değerler arasındaki uzaklığın bulunması gereklidir. Buradaki uzaklık ölçütü öklid uzaklığı olup formülü şu şekildedir.

$$\text{öklid}(a,b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Bu problemde toplam 10 ölçüm olduğu için toplamda bütün noktaların birbirine olan uzaklığını bulup 10x10'luk bir matrise atmamız gereklidir.

Burada ben sadece en kısa kenarların değerlerini matrise yazacağım (normalde hepsi hesaplanır)

	a 1	b 2	c 3	d 4	e 5	f 6	g 7	h 8	i 9	j 10
a 1	0					1			1	
b 2		0					$\sqrt{5}$	$\sqrt{5}$		
c 3			0				$\sqrt{8}$			
d 4				0					$\sqrt{5}$	3
e 5					0	1				
f 6	1				1	0				
g 7		$\sqrt{5}$	$\sqrt{8}$				0			
h 8		$\sqrt{5}$						0		$\sqrt{20}$
i 9	1			$\sqrt{5}$					0	
j 10				3				$\sqrt{20}$		0

öklit(i, j) = öklit(j, i) = $\sqrt{(17-19)^2+(8-10)^2} = \sqrt{8}$

öklit(e, f) = öklit(f, e) = $\sqrt{(9-9)^2+(21-20)^2} = 1$

öklit(a, f) = öklit(f, a) = $\sqrt{(10-9)^2+(20-20)^2} = 1$

öklit(a, i) = öklit(i, a) = $\sqrt{(10-10)^2+(20-19)^2} = 1$

öklit(j, d) = öklit(d, j) = $\sqrt{(8-11)^2+(17-17)^2} = 3$

öklit(i, d) = öklit(d, i) = $\sqrt{(10-11)^2+(19-17)^2} = \sqrt{5}$

öklit(j, h) = öklit(h, j) = $\sqrt{(13-17)^2+(6-8)^2} = \sqrt{20}$

öklit(b, h) = öklit(h, b) = $\sqrt{(12-13)^2+(4-6)^2} = \sqrt{5}$

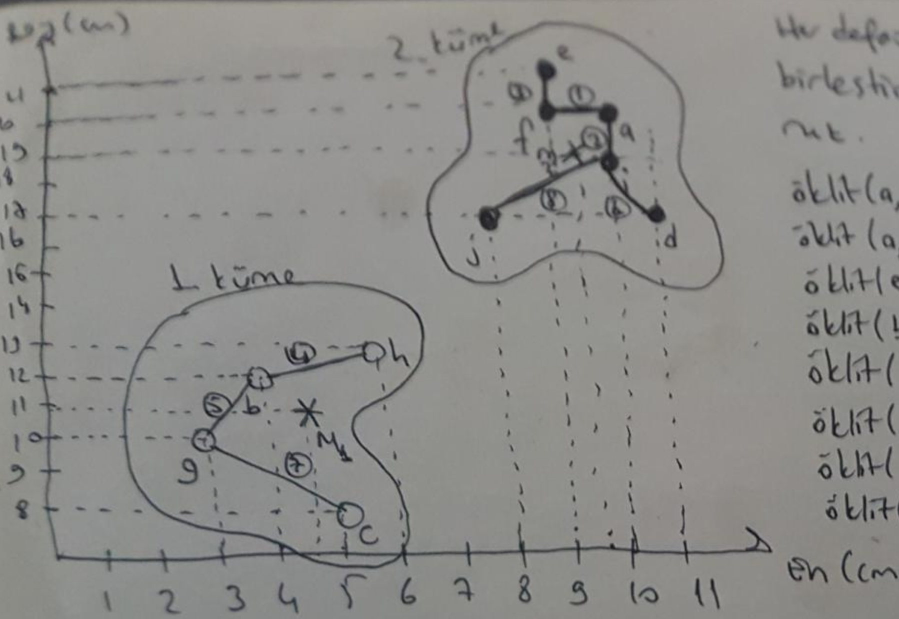
öklit(b, g) = öklit(g, b) = $\sqrt{(10-12)^2+(3-4)^2} = \sqrt{5}$

öklit(g, c) = öklit(c, g) = $\sqrt{(10-12)^2+(3-5)^2} = \sqrt{8}$

Kruskal algoritması ile kümeleme yaparken diğer bir belirlemesi gereken şey küme sayısıdır. Biz burada küme sayısını 2 alalım.

Kruskal algoritmasının mantığında minimum spanning tree(MST)leri birbirine en kısa kenarla birleştirerek yeni MST'ler oluşturmak vardı. En son aşamada iki MST birleştirildiğinde sonuç elde ediliyordu. Kümeleme yapılırkense bu birleştirme işlemleri yapılırken her defasında kaç tane minimum spanning tree kaldığı kontrol edilmelidir. Eğer 2 kümeye böleceksek son olarak 2 tane MST kalmalıdır. Her MST'ye ait noktalar iki farklı kümenin elemanları olacaktır.

Şimdi ise grafik üzerinden MST'leri oluşturarak sonucu elde edelim.



Her defasında en kısa kenarı birleştirecek MST 'leri birleştiriyoruz.

öklit(a,f) = 1 (1. adım)
öklit (a,i) = 1 (2. adım)
öklit(e,f) = 1 (3. adım)
öklit(b,h) = $\sqrt{5}$ (4. adım)
öklit(b,g) = $\sqrt{5}$ (5. adım)
öklit(i,d) = $\sqrt{5}$ (6. adım)
öklit(g,c) = $\sqrt{8}$ (7. adım)
öklit(i,j) = $\sqrt{8}$ (8. adım)

Görüldüğü gibi 2 MST kaldığında 2 tane kümeye ayırmış olduk 1. kümenin üyeleri {h,b,g,c} ikinci kümenin üyeleri {a,e,d,i,f,j} oldu. Bu kümelerin orta noktaları m_1, m_2 ise küme elemanların x,y değelerinin ortalaması olur.

$m_1 = \left(\frac{6+4+3+5}{4}, \frac{13+12+10+8}{4}\right) = (4.5, 10.75)$

$m_2 = \left(\frac{11+10+10+9+9+8}{6}, \frac{17+17+19+20+[illegible]+21}{6}\right)$

$m_2 = (9.5, 19)$