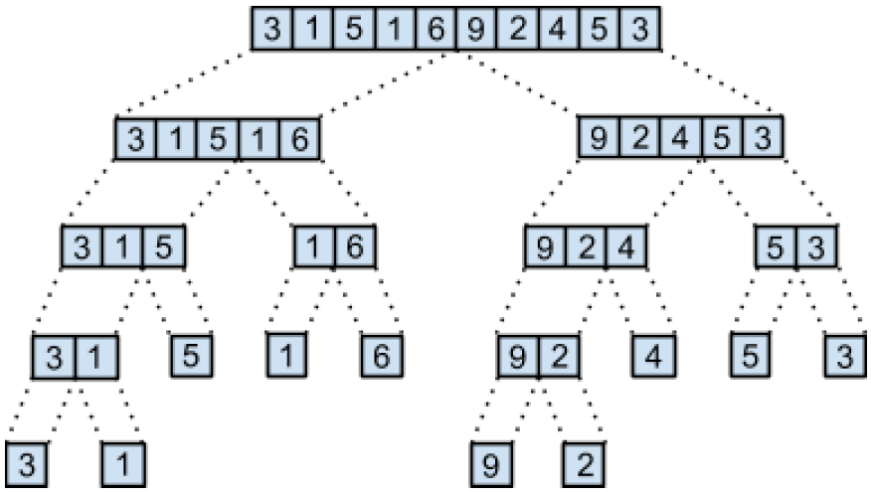
Mergesort

Este algoritmo utiliza la técnica de divide y vencerás para ordenar un vector *v*. El vector *v* se divide en dos sub-vectores de igual tamaño y se ordena de forma recursiva cada uno de ellos por separado. Las llamadas recursivas se detienen cuando el tamaño del sub-vector es 1, que es precisamente cuando el sub-vector está ordenado.

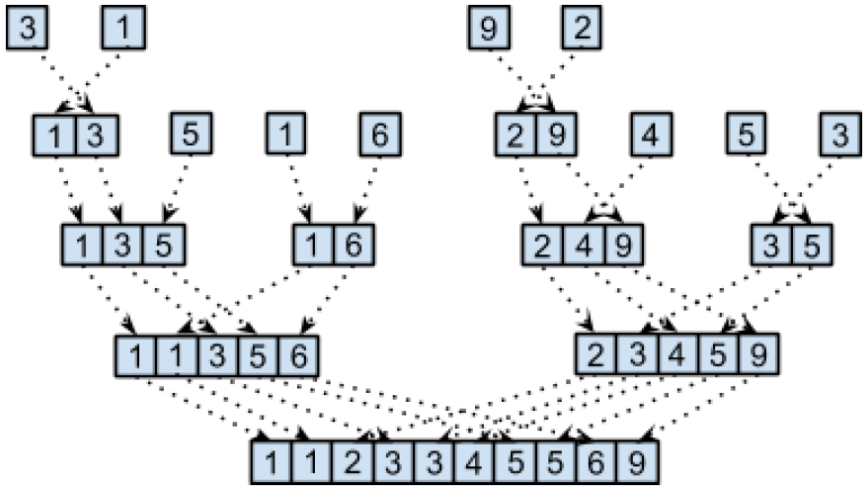
Posteriormente, para construir el vector original de tamaño n, se fusionan los dos sub-vectores ordenados.

Veamos una demostración con un ejemplo:

El vector se divide en dos y se hace la llamada recursiva sobre cada mitad.



Cuando las divisiones dan lugar a sub-vectores de tamaño 1, éstos se fusionan de forma ordenada, como se muestra a continuación



Finalmente, obtenemos el vector de tamaño n ordenado.

Algoritmo mergesort:

```
void Mergesort(int v[], int inicio, int final){
	int centro;
	if (inicio < final) {
		centro=(inicio+final)/2;
		Mergesort(v,inicio,centro);
		Mergesort(v,centro+1,final);
		Fusion(v,inicio,centro+1,final);
	}
}

void Fusion(int v[], int inicio, int centro, int final){
	int aux[final – inicio + 1];
	int h, i, j, k;
	h = inicio;
	i = incio;
	j = centro+1;
	while ((i <= centro) && (j <= final)) {
		if (v[i] <= v[j]) {
			aux[h] = v[i];
			i++;
		}else{
			aux[h] = v[j];
			j++;
		}
		h++;
	}
//Si se copiaron todos los elementos de la primera mitad, se copia el resto de la segunda mitad.
	If ( i > centro ) {
		for (k= j; k <= fin; k++) {
			aux[h]=v[k];
			h++;
		}
//sino se copia el resto de la primera mitad
	}else{
		for (k= i; k <= centro; k++) {
			aux[h]=v[k];
			h++;
		}
	}
//Finalmente se copia el contenido al vector original
	for (k= inicio; k <= fin; k++)
		v[k]=aux[k];
}
```