

```

// min returnerar det minsta elementet i en sekventiell samling.
// Om samlingen är tom, kastas ett undantag av typen IllegalArgumentException.
public static int min (int[] element) throws IllegalArgumentException
{
    if (element.length == 0)
        throw new IllegalArgumentException ("tom samling");

    // hör ihop med spårutskriften 2:
    // int antalVarv = 1;

    int[] sekvens = element;
    int antaletPar = sekvens.length / 2; dela i par
    int antaletOparadeElement = sekvens.length % 2; om sekvens är inte jämnta
    int antaletTankbaraElement = antaletPar + antaletOparadeElement; (par)+(par)+(ensam)
    int[] delsekvens = new int[antaletTankbaraElement];
    int i = 0;
    int j = 0;
    while (sekvens.length > 1)
    {
        // skilj ur en delsekvens med de tänkbara elementen
        i = 0;
        j = 0;
        while (j < antaletPar); kör flera varv
        {  

            delsekvens[j++] = (sekvens[i] < sekvens[i + 1])? sekvens[i] : sekvens[i + 1]; skapa plats för att lägga alla element  

            i += 2; T0J>T2J>T4J . . .
        }  

        if (antaletOparadeElement == 1)  

            delsekvens[j] = sekvens[sekvens.length - 1]; sista ensam element hålla kvar.  

        // utgå nu ifrån delsekvensen  

        sekvens = delsekvens; Antal par + ensam  

        antaletPar = antaletTankbaraElement / 2; dela i par igen  

        antaletOparadeElement = antaletTankbaraElement % 2; ensamtal eller inte ensam  

        antaletTankbaraElement = antaletPar + antaletOparadeElement; par+par+ensam  

* Fel = delsekvens = new int[antaletTankbaraElement] har inte skapa plats för nästa varv.  

        // spårutskrift 1 - för att följa sekvensen  

        // System.out.println (java.util.Arrays.toString (sekvens));
    }

    // sekvens[0] är det enda återstående tänkbara elementet
    // - det är det minsta elementet
    return sekvens[0];
}

```

1.. Spåra metoden `min` i fallet att sekvensen innehåller sexton element. Använd papper och penna: rita en serie bilder som visar hur talsekvensen transformeras. Analysera transformationer av talsekvensen, och hitta det första felet i metoden. Rätta detta fel.

$$\text{Sekvens} = 16$$

$$\text{antaletPar} = 8$$

$$\text{antaletOparadeElement} = 0$$

$$\text{antaletTankbaraElement} = 8$$

$$\text{delsekvens} = \text{int}[8]$$

par:	i	i+1
------	---	-----

*(jämförelse kör i par)*

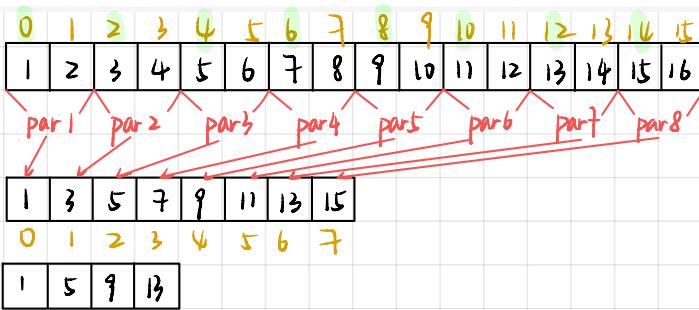
*om delsekvens[i] = minst*

$\Rightarrow \text{delsekvens}[i] = \text{delsekvens}[i]$

*om delsekvens[i+1] = minst*

$\Rightarrow \text{delsekvens}[i+1] = \text{delsekvens}[i+1]$

*\* Fel = delsekvens = new int[antaletTankbaraElement]* *har inte skapa plats för nästa varv.*



1	9
---	---

1
---

```

// min returnerar det minsta elementet i en sekventiell samling.
// Om samlingen är tom, kastas ett undantag av typen IllegalArgumentException.
public static int min (int[] element) throws IllegalArgumentException
{
    if (element.length == 0)
        throw new IllegalArgumentException ("tom samling");

    // här ihop med spårutskriften 2:
    // int antalVarv = 1;

    int[] sekvens = element;
    int antaletPar = sekvens.length / 2;
    int antaletOparadeElement = sekvens.length % 2;
    int antaletTankbaraElement = antaletPar + antaletOparadeElement;
    int[] delsekvens = new int[antaletTankbaraElement];
    int i = 0;
    int j = 0;
    while (sekvens.length > 1)
    {
        // skilj ur en delsekvens med de tankbara elementen
        i = 0;
        j = 0;
        while (j < antaletPar)
        {
            delsekvens[j++] = (sekvens[i] < sekvens[i + 1])? sekvens[i] : sekvens[i + 1];
            i += 2;
        }
        if (antaletOparadeElement == 1)
            delsekvens[j] = sekvens[sekvens.length - 1];

        // utgå nu ifrån delsekvensen
        sekvens = delsekvens;
        antaletPar = antaletTankbaraElement / 2;
        antaletOparadeElement = antaletTankbaraElement % 2;
        antaletTankbaraElement = antaletPar + antaletOparadeElement;
        delsekvens = new int[antaletTankbaraElement];
        // spårutskrift 1 - för att följa sekvensen
        // System.out.println (java.util.Arrays.toString (sekvens));

        // spårutskrift 2 - för att avsluta loopen i förväg
        // (för att kunna se vad som händer i början)
        // if (antalVarv++ == 10)
        //     System.exit (0);
    }

    // sekvens[0] är det enda återstående tankbara elementet
    // - det är det minsta elementet
    return sekvens[0];
}

```

2. För att hitta **det andra felet** spåra den korrigerade varianten av metoden `min`. Använd en sekvens som innehåller nitton element med det minsta elementet på den sjuttonde positionen. Rita en serie bilder som visar hur det minsta helalet bestäms. Varför erhålls ett felaktigt svar? Hitta det andra felet i metoden och rätta det.

