EU2

2.

Elementjämnförelser av sortera(n, X) blir antalet element *n* i kvadrat då en loop genom alla element själv innehåller en loop genom resterande element.

Komplexitetsfunktion: n2

θ(n2)

3.

Bästa (Elementen är redan sorterade) = Inga elementutbyten - θ(n)

Värsta (Utbyte varje jämförelse, sorterad ”åt andra hållet”) = n2 elementutbyten - θ(n2)

Medel (Utbyte varannan jämförelse) = n2 elementutbyten - θ(n2)

4.

Jämnförelse:

* Utbytessortering θ(n2)
* Urvalssortering θ(n2)

Utbyten:

* Bästa:
  + Utbytessortering θ(n)
  + Urvalssortering θ(n2)
* Värsta:
  + Utbytessortering θ(n2)
  + Urvalssortering θ(n2)
* Medel:
  + Utbytessortering θ(n2)
  + Urvalssortering θ(n2)

5.

A)

**INRE LOOPEN**

**ETT PÅSTÅENDE OM INRE LOOPEN**

När den inre loopen har utförts, gäller följande:

xi = minimum {xi, xi+1, …, xn}

**BEVIS**

Exempel:

*i* = 0

*j* = 1

*xi* -> *xn* = {6,2,5,3,1}

*n* = 5

I detta fall kommer xi alltid vara första elementet i listan och xj kommer för varje loop i inre loopen ta värdet av nästa element till höger, startandes på 2. Vid varje loop jämförs xi och xj, om xj är mindre än xi kommer värden byta plats i listan så att elementet på xi kommer vara den minsta av de två.

Således kommer xi vara minsta värdet (minimum) av alla element i listan xi -> xn.

B)

HUVUDLOOPEN

ETT PÅSTÅENDE OM HUVUDLOOPEN

När huvudloopen har utförts, gäller följande:

x1 < x2 < … < xn

BEVIS

Exempel:

*i* = 0

*xi* -> *xn* = {6,2,5,3,1}

*n* = 5

I varje loop av huvudloopen kommer xi sättas till minsta värdet av *xi* -> *xn* med hjälp av inre loopen. *i* ökas sedan med ett så att *i* representerar nästa element i listan och i nästa loop sätts den nya xi till det minsta värdet av de resterade elementen i listan. Denna loop fortsätter tills i är lika med n och då har eftervillkoret uppfyllts.