2. Antal jämförelseoperationer:

Funktionen t(n) = (n-1) + (n-2) + (n -3) + … + 3 + 2 + 1

t(n) = n(n-1)/2

t(n)  θ(n2)

Funktionen är en del av θ-mängden θ(n2)

3.

(Sannolikheten för ett utbyte i bästa fall är 0)

Det bästa fallet för sorteringsalgoritmen är när vektorn redan är sorterad vilket innebär att den ej kommer behöva göra några utbyten b(n) = θ(1)

(Sannolikheten för ett utbyte i värsta fall är 1)

Det värsta fallet för sorteringsalgoritmen är när vektorn är sorterad från högsta till minsta. Detta innebär att ett utbyte kommer ske för jämförelse. Eftersom algoritmen kommer att passera båda looparna n gånger innebär det w(n) = θ(n2).

(Sannolikheten för ett utbyte i ett genomsnittligt fall är 1/2)

Medelfallet är precis som det värsta fallet då algoritmen kommer behöva passera båda looparna n gånger vilket innebär n\*n som är a(n) θ(n2).

1. INRE LOOPEN

ETT PÅSTÅENDE OM INRE LOOPEN

När den inre loopen har utförts, gäller följande:

xi = minimum {xi, xi+1, …, xn}

BEVIS

Variabel j = i + 1

Algoritmen jämför det första värdet i elementet (X[i] med nästkommande element (X[j]). Om nästkommande element har ett mindre värde så ersätts första värdet med nästkommande element (X[i] = X[j]). Det nu ersatta första värdet X[j] jämförs med nu tredje elementet (X[j+1]). Detta görs framtill att variabeln j har samma värde som det finns antal element.

När denna loop har gjorts så kommer det minsta elementet i mängden vara på första platsen då alla element i mängden kommer att jämföras med det nuvarande minsta.

En ordnad mängd X innehåller n olika element. Där n är ett naturligt tal mer än eller lika med 1

X = {x1,x2,x3,…xn-1,xn}

Om inga av elementen är mindre än x1 så är x1 minst av alla element.

(x1< x2 x1 < x2  x1 < x3  x1 < xn-1  x1 < xn) ( x1< xn) x1= xi

För varje iteration i den inre loopen, jämförs alla element efter och från index i+1 med det element på index i. Om de är mindre än det element som för tillfället finns på index i, byter de plats och fortsätter med nästa tal. På så sätt kommer alltid det minsta talet ligga på position i efter att den inre loopen har utförts.

1. HUVUDLOOPEN ETT PÅSTÅENDE OM HUVUDLOOPEN

När huvudloopen har utförts, gäller följande: x1 < x2 < … < xn

BEVIS

Om den inre loopen alltid slutar med att det minsta talet av alla tal efter i mängden, ligger på position i, så måste alla tal ligga i storleksordning ifall man utför innerloopen på varje tal index i mängden från vänster till höger (från 0 till mängdens längd) då minsta talet kommer läggas först på index 0, sen näst största på index 1 (minsta av de resesterande talen), osv..