

Searching and sorting

Linear Search

1 [Initialize the search]

$i=1$

2 [enter the value]

enter value of x

3 [search into the array]

Repeat while $a[i] \neq x$

$i=i+1$

4[successful search?]

if $i == n$

write("unsuccessful")

else

write("successful")

return(i)

Binary search

1 [initialize]

low=1, high=n

2 [Perform search]

Repeat thru step 4 while low<=high

3 [get the index of midpoint]

mid=(low+high)/2

4 [compare]

if $x < a[\text{mid}]$

 high=mid-1

else if $x > a[\text{mid}]$

 low=mid+1

else if $x == a[\text{mid}]$

 return (mid)

Bubble sort

1 [initialize]

last = n

2 [loop outer for pass count]

Repeat thru step 5 for $i=1,2,\dots,n-1$

3 [exchange count]

exch = 0

4 [perform pairwise comparison]

repeat for $j=1,2,3,\dots,last-1$

if $k[j] < k[j+1]$

swap $k[j]$ and $k[j+1]$

exch++

5 [abt exchange on this pass?]

if exch=0

return

else

last=last-1

Selection sort

1 [initialize]

initialize an array

2 [loop]

repeat thru step 5 for $i=1,2,\dots,n$

3 [initialize min index]

$\text{min_index}=i$

4 [do comparisons and find smallest value]

repeat for $j=i+1,i+2,\dots,n$

if $k[j] < k[\text{min_index}]$

$\text{min_index} = j$

5 [exchange]

if $\text{min_index} \neq i$

swap $k[i]$ and $k[\text{min_index}]$

Quicksort(a,lb,ub)

1 [Initialize]

flag=true

2 [perform sort]

if lb < ub

 i= lb, j=ub, pivot = a[lb]

while (i <j)

 repeat while a[i] < pivot

 i=i+1

 repeat while a[j] > pivot

 j=j-1

 if i < j

 swap a[i] and a[j]

 else

 flag = false

swap a[j] and pivot

call quicksort(a,lb,j-1)

call quicksort(a,j+1,ub)

Mergesort(a,l,r)

1 [find the middle point to divide array]

$$m = (l+r)/2$$

2 [call mergesort for first half]

call mergesort(a,l,m)

3 [call mergesort for second half]

call mergesort(a,m+1,r)

4 [merge]

call merge(a,l,m,r)

Merge(a,first,second,third)

1 [initialize]

i=first, j=second, l=0

2 [compare and output smallest]

repeat while i<second and j<=third

if a[i] <= a[j]

l=l+1

temp[l] = a[i]

i = i + 1

else l=l+1

temp[l] = a[j]

j = j + 1

3 [copy remaining unprocessed elements]

if i >= second

repeat while j <= third

l=l+1

temp[l] = a[j]

j = j + 1

else repeat while i < second

l=l+1

temp[l] = a[i]

i = i + 1

4 [display output]