

Informatics

Decision Trees

Francisco Iacobelli

Department of Computer Science
Northeastern Illinois University

September 13, 2011

Sorting

- Rank links, songs, etc.
- Optimize searches (Last name, neuid, etc)

Bubble Sort

BubbleSort (list)

```
repeat
  for each element of the list
    compare the next adjacent element
    if the elements are in the wrong order: swap them
until no swaps are needed
```

Example: Sort {5, 3, 1, 6, 8, 4, 7, 2}

BubbleSort

Pseudocode

```
procedure bubbleSort( A : list)
repeat
  swapped = false
  for i = 1 to length(A) - 1 do:
    if A[i-1] > A[i] then
      swap( A[i-1], A[i] )
      swapped = true
    end if
  end for
until not swapped
end procedure
```

Selection Sort

Find the minimum value in the list
Swap it with the value in the first position
Repeat the steps above for the remainder of the

Example: Sort {5, 3, 1, 6, 8, 4, 7, 2}

Notice how the list is split into sorted and unsorted elements

Selection Sort

```
pos = 0
min = 0
for pos = 0 to length(A) - 1 do
    min = pos
    for i = pos+1 to length(A) -1 do
        if A[i] < A[min])
            min = i;
    if min != pos
        swap(A[pos], A[min]);
```

Insertion Sort

```
repeat
  remove an element from the input data
  insert it into the correct position to the left
until no input elements remain.
```

Example: Sort {5, 3, 1, 6, 8, 4, 7, 2}

InsertionSort

Pseudocode

```
for j = 1 to length(A)-1
  key = A[ j ]
  i = j - 1
  while i >= 0 and A [ i ] > key do
    A[ i +1 ] = A[ i ]
    i = i -1
  A [i +1] = key
```