Algorithms:

Def: An algorithm is a finite Set of Steps for performing a computation

Ex: Find the maximum element of $\{1, 7, 2, 31, 14, 17\} = l$

In words what are the Steps?

Set max = 1 (first element)

Compare with each new element & if bigger set max to it.

Stop when out of terms.

In pseudo code:

```
max = l[0]
for i = 1 ... n :
    if l[i] > max
        max = l[i]
return max.
```

all algorithms must have certain properties:

Input: values given to the algorithm

output: Values returned by algorithm

Definiteness: Each step of the algorithm must be completely defined. (can't have statements like 2. find the square of (n))

Correctness: Algorithms Should produce the correct output always.

Finiteness: Algorithm must terminate. In some amount of finite time (possibly large) the algorithm must give its output.

Effectiveness ; Each step of the algorithm must be computable.

Generality : An algorithm must solve all problems of the desired form.
e.g. doesn't find max{1, 7, 2, 31, 14, 17} but any list.

Searching algorithms :

Sequential Search : Given a value & a list check each element of list against yours.

Sometimes you can assume list is sorted.

```
linear-search (x, l) :
    i=1
    while ((i≤len(l)) & (x ≠ l[i]))
        i++;
    if i≤len(l)
        return i
    else
        return -1,
```

```
e.g. x=5 l=[1, 2, 4, 5, 7, 9]
  i=1    len(l)=6
      5≠1
  i=2
      5≠2
  i=3
      5≠4
  i=4
      5=5 ✓
      return 4
```

Binary Search : This requires the list we're searching to be in order.

Idea ; given x, check mid value of list, if its x done, if its smaller than x then examine list after mid point if its bigger than x examine list before mid point.

Ex $l = [1, 2, 3, 5, 6, 7, 8, 10, 12, 13, 15, 16, 18, 19, 20, 22]$

We search this list for 15. (there are 16 terms) ~~[illegible]~~

first we split into two lists of 8

1, 2, 3, 5, 6, 7, 8, 10   12, 13, 15, 16, 18, 19, 20, 22

Compare largest element of first list to our element;

$10 < 15 \Rightarrow$ 15 (if it is in list) must be in other list.

so split that into 2 lists of 4.

12, 13, 15, 16   18, 19, 20, 22

$16 > 15$ so 15 (if it exists must be in the first list

Split this into two lists of 2

12, 13   15, 16

$13 < 15 \Rightarrow$ in other list

15   16

$15 = 15$ found it!

Algorithm:

```
                          sorted list
binary Search (x, l )

i    i = 1
     j = n = len(l)

     while i < j :
          m = $\lfloor \frac{(i+j)}{2} \rfloor$

          if x > l[m]
                    i = m+1
          else
                j = m

     if x = l[i]
          return i

     else
          return -1.
```

Could also talk about Sorting, but you'll do that in algorithms.

Lets do greedy algs. These are algorithms where the best option at each step is always taken.

Ex: How can you make 67 cents from American Coins; with the least amount of Coins?

Greedily you'd take the most cents per coin each step:

2-25 cent Coins, 1-10 cent coin 1-5 cent coin 2-1 cent coins.