CMPS 10 10-13-09

To Determine Run Time of An Algorithm We:

- Choose a <u>Basic Operation</u> to Count.
- For a fixed input size n, Count # of Basic Ops in <u>Best</u>, <u>Worst</u>, And <u>Average</u> Cases.

Ex. Recall Seq. Search.

```
1.) get n, a_1, ..., a_n, target
2.) i ← 1
3.) found ← false
4.) while i ≤ n and not found
5.)     [if (a_i = target)
6.)         [found ← true
7.)     else
8.)         [i ← i + 1
9.) if not found
10.)    [i ← 0
11.) Print i
12.) stop
```

BASIC OP: Comparison of target with an element in list i.e. line 5.

Best Case: 1 COMPARISON

WORST CASE: n COMP.

AVG. CASE:

- Assume target is in list.
- Assume target is equally likely to be at any position in list.

if target in loc	# comp
1	1
2	2
3	3
⋮	⋮
n	n

$$\text{avg \# comp} = \frac{1+2+3+\cdots+n}{n} = \frac{\frac{n(n+1)}{2}}{n} = \frac{1}{2}n + \frac{1}{2}$$

Summarize:

	# comp
Best	1
worst	n
avg	$\frac{1}{2}n + \frac{1}{2}$

Exercise:

Calculate avg. case # comp. under the Assumption that target is EQually likely to be in list or not, and if target is in list, it is equally likely to be in any Pos. in list

Answer: $\frac{3n+1}{4}$

Sorting

Re-arrange an unordered list into increasing order.

Selection Sort:

Input: $n \geq 1$ (length of list to be sorted)

$a_1, a_2, \ldots, a_n$ (the list itself)

Output: original list in increasing order.