CSE 373 / L-02 / 11.02.2024 /

Algorithm:

- procedure
- input
- output
- time

A sequence of ~~time~~ computational steps that transform the input into output.

Insertion Sort:

⇒ Data will be sort during insertion

⇒ The number to be sorted are known as keys.

⇒ Algorithm

```
1  for i = 2 to n
2      key = A[i]
3  //Comment
4      j = i - 1
5      while j > 0 and A[j] > key
6          A[j+1] = A[j]
7          j = j - 1
8      A[j+1] = key
```

Code:

```
for i in range (2, n):
    key = A[i]
    j = i - 1
    while j > 0 && A[j] > key:
        A[j+1] = A[j]
        j = j - 1
    A[j+1] = key
```

Random Access Machine (RAM)

- Predicting the resources that algorithm requires.
- computational time that we want to measure

RAM:

- one processor model
- Instruction are executed one after another
- instruction - operation takes a constant amount of time.
- Data type - integer, float, character
- No memory hierarchy

Running Time:

	cost	times
1 ⇒	c_1	n
2 ⇒	c_2	$n-1$
3 ⇒	0	$n-1$
4 ⇒	c_4	$n-1$
5 ⇒	c_5	$\sum_{i=2}^{n} t_i$
6 ⇒	c_6	$\sum_{i=2}^{n} (t_i-1)$
7 ⇒	c_7	$\sum_{i=2}^{n} (t_i-1)$
8 ⇒	c_8	$n-1$

Running time!

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5 \sum_{i=2}^{n} t_i + c_6 \sum_{i=2}^{n} (t_i - 1) + c_7 \sum_{i=2}^{n} (t_i - 1) + c_8(n-1)$$

⇒ Best Case!

→ while statement will false always

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5(n-1) + c_8(n-1)$$

$$= \underbrace{(c_1 + c_2 + c_4 + c_5 + c_8)}_{a} n - \underbrace{(c_2 + c_4 + c_5 + c_8)}_{b}$$

$$= an + b \Rightarrow \text{linear function}$$

⇒ Worst Case (gives upper bound - maximum of time)

$t_i = i$ (follow case d on slide)

$$\sum_{i=2}^{n} i = \left(\sum_{i=1}^{n} i\right) - 1 = \frac{n(n+1)}{2} - 1$$

$$\sum_{i=2}^{n} (i-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2}$$

$$T(n) = c_1 n + c_2(n-1) + c_4(n-1) + c_5\left(\frac{n(n+1)}{2} - 1\right) + c_6\left(\frac{n(n-1)}{2}\right) + c_7\left(\frac{n(n-1)}{2}\right) + c_8(n-1)$$

$$= \underbrace{\left(\frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2}\right)}_{a} n^2 + \underbrace{\left(c_1 + c_2 + c_4 + \frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2} + c_8\right)}_{b} n - \underbrace{(c_2 + c_4 + c_5 + c_8)}_{c}$$

$$= an^2 + bn + c \Rightarrow \text{quadratic function}$$

Order of Growth / order of growth

- consider only the leading term of a formula
  - $an^2 + bn + c \Rightarrow an^2$
- Ignore the leading term's constant coefficient
  - $an^2 \Rightarrow n^2$

⇒ Def^n: We ignore the actual cost of each statement, using the constants c_i to represent these costs.

⇒ Worst case running time of insertion sort is $\Theta(n^2)$

We usually consider one algorithm to be more efficient than another if its worst-case running time has a lower order of growth.