## Introduction

We find the time complexity of several common algorithms in this lecture including finding the maximum element in a finite sequence, the linear search algorithm, the binary search algorithm, the bubble sort algorithm, and the insertion sort algorithm,

**Algorithm 1: Finding the Maximum Element in a Finite Sequence**

| Line | Code | Cost |
|---|---|---|
| 1 | **int** *Max*(**int** *A*[], **int** *n*) | $0$ |
| 2 | { **int** *m*=*INT_MIN*; | $1$ |
| 3 | **int** *a*=0; | $1$ |
| 4 | **while** (*a*<*n*) { | $n = \sum_{a=0}^{n-1} 1$ |
| 5 | **if** (*A*[*a*]>*m*) *m*=*A*[*a*]; | $2n$ |
| | *a*++; | $n$ |
| 6 | } | $1$ |
| 7 | **return** m; | $0$ |
| 8 | } | $0$ |
| **Total** | | $T(n) = 4n + 3$ |

1. $f(n) = n$
2. $C = 4 + 1 = 5$
3. $|T(n_0)| \le Cf(n_0)$
   $|4n_0 + 3| \le |5n_0|$
   $n_0 \ge 3$

$T(n)$ is $O(n)$ because we found witnesses $C = 5$ and $n_0 = 3$ that make $|T(n_0)| \le Cf(n_0)$ for $n \ge n_0$

**Algorithm 2: The Linear Search Algorithm**

| Line | Code | Cost |
|---|---|---|
| 1 | **int** *LinearSearch*(*A*[], **int** *n*,**int** *key*) | $0$ |
| 2 | { **int** *ndx*=-1; | $1$ |
| 3 | **int** *a*=0; | $1$ |
| 4 | **while** (*a*<*n*) { | $n = \sum_{a=0}^{n-1} 1$ |
| 5 | **if** (*A*[*a*]==*key*) *ndx*=*a*; | $2n$ |
| 6 | *a*++; | $n$ |
| 7 | } | $1$ |
| 8 | **return** *ndx*; | $0$ |
| 9 | } | $0$ |
| **Total** | | $T(n) = 4n + 3$ |

1. $f(n) = n$
2. $C = 4 + 1 = 5$
3. $|T(n_0)| \leq Cf(n_0)$
   $|4n_0 + 3| \leq |5n_0|$
   $n_0 \geq 3$

$T(n)$ **is** $O(n)$ **because we found witnesses** $C = 5$ **and** $n_0 = 3$ **that make** $|T(n_0)| \leq Cf(n_0)$ **for** $n \geq n_0$

**Algorithm 3: The Binary Search Algorithm**

| Code | Cost |
|---|---|
| **int** *BinarySearch*(*A*[], **int** *n*,**int** *key*) | |
| { **int** *lo*=**0;** | $1$ |
| **int** *hi*=*n*-**1;** | $1$ |
| **while (***lo*<=*hi***) {** | $k$ |
| **int** *m*=(*hi*+*lo*)**/2;** | $3k$ |
| **if (***key*==*A*[*m*]**) return** *m*; | $k$ |
| **if (***key*<*A*[*m*]**)** | $k$ |
| *hi*=*m*-**1;** | $2k$ |
| **else** | |
| *lo*=*m*+**1;** | $2k$ |
| **}** | $1$ |
| **return -1;** | |
| **}** | |
| **Total** | $\boldsymbol{T(n) = 8k + 3}$ |
| | $\boldsymbol{T(n) = 8\lceil \log_2 n \rceil + 3}$ |

Analysis:
Goal: find $k = f(n)$

1. $n_i = \frac{n_{i-1}}{2}$
2. $n_0 = n$, the initial value of $n$ – the number of values to search.
3. $n_k = 1$, In the $k$th iteration of the loop, when *hi*=*lo*, there is only one value to search.
4. $n_i = \frac{n}{2^i}$
5. $n_k = \frac{n}{2^k} = 1$
6. From 5, $n = 2^k \Longrightarrow k = \log_2 n$
7. Assume the worst case: $k = \lceil \log_2 n \rceil$

**Finding** $\boldsymbol{O\big(f(n)\big)}$:

1. $\boldsymbol{f(n) = \lceil \log_2 n \rceil}$
2. $\boldsymbol{C = 8 + 1 = 9}$
3. $\boldsymbol{|T(n_0)| \le C|f(n_0)|}$
   $\boldsymbol{|8\lceil \log_2 n_0 \rceil + 3| \le 9|\lceil \log_2 n_0 \rceil|}$
   $\boldsymbol{\log_2 n_0 \ge 3}$
   $\boldsymbol{n_0 \ge 2^3}$
   $\boldsymbol{n_0 \ge 8}$

$\boldsymbol{T(n)}$ **is** $\boldsymbol{O(\log_2 n)}$ **because we found witnesses** $\boldsymbol{C = 9}$ **and** $\boldsymbol{n_0 = 8}$ **that make** $\boldsymbol{|T(n_0)| \le Cf(n_0)}$ **for** $\boldsymbol{n \ge n_0}$

**Algorithm 4: The Bubble Sort**

| Line | Code | Cost |
|---|---|---|
| 1 | **void** *BubbleSort*(**int** *A*[], **int** *n*) | |
| 2 | { **int** *flag*=1; | $1$ |
| 3 | **int** *i*=1; | $1$ |
| 4 | **while** (*i*<=*n*) { | $a = \sum_{i=1}^{n} 1$ |
| 5 | *flag*=0; | $a$ |
| 6 | **int** *j*=0; | $a$ |
| 7 | **while** (*j*<(*n*-1)) { | $b = \sum_{i=1}^{n} \sum_{j=0}^{n-2} 1, 2b$ |
| 8 | **if** (*A*[*j*+1]>*A*[*j*]) { | $2b$ |
| 9 | **int** *temp*=*A*[*j*]; | $b$ |
| 10 | *A*[*j*]=*A*[*j*+1]; | $2b$ |
| 11 | *A*[*j*+1]=*temp*; | $2b$ |
| 12 | *flag*=1; | $b$ |
| 13 | } | |
| 14 | *j*++; | $b$ |
| 15 | } | $b$ |
| 16 | *i*++; | $a$ |
| 17 | } | $1$ |
| 18 | } | |
| | $b = \sum_{i=1}^{n} \sum_{j=0}^{n-2} 1$<br>$b = \sum_{i=1}^{n} (n-1)$<br>$b = n(n-1) = n^2 - n$ | |
| | $a = \sum_{i=1}^{n} 1 = n$ | |
| | $T(n) = 12b + 4a + 3 = 12(n^2 - n) + 4n + 3$ | |
| | $\mathbf{T(n) = 12n^2 - 8n + 3}$ | |

**Finding** $O(f(n))$:

1. $f(n) = n^2$
2. $C = 12 + 1 = 13$
3. $|T(n_0)| \le C|f(n_0)|$

   $|12n_0^2 - 8n_0 + 3| \le 13|n_0^2|$

   $n_0^2 + 8n_0 - 3 \ge 0$

   $n_0 = \left\lceil \frac{-8 \pm \sqrt{8^2 + 4 \cdot 3}}{2} \right\rceil = 1$

**$T(n)$ is $O(n^2)$ because we found witnesses $C = 13$ and $n_0 = 1$ that make $|T(n_0)| \le Cf(n_0)$ for $n \ge n_0$**

**Algorithm 5: The Insertion Sort**

| Line | Code | Cost |
|---|---|---|
| 1 | **void** *Insertion_Sort*(**int** *A*[],**int** *n*) | |
| 2 | { **int** *i*=1; | $1$ |
| 3 | **while** (*i*<*n*) { | $a = \sum_{i=1}^{n-1} 1$ |
| 4 | **int** *j*=*i*; | $a$ |
| 5 | **while** (*j*>0 && *A*[*j*-1]>*A*[*j*]) { | $b = \sum_{i=1}^{n-1} \sum_{j=1}^{i} 1, 4b$ |
| 6 | **int** *t*=*A*[*j*]; | $b$ |
| 7 | *A*[*j*]=*A*[*j*-1]; | $2b$ |
| 8 | *A*[*j-1*]=*t*; | $2b$ |
| 9 | *j*--; | $b$ |
| 10 | } | $a$ |
| 11 | *i*++; | $a$ |
| 12 | } | $1$ |
| 13 | } | |

$$T(n) = 10b + 4a + 2$$

$b = \sum_{i=1}^{n-1} \sum_{j=1}^{i} 1$

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

$b = \sum_{i=1}^{n-1} i - \sum_{i=1}^{n-1} 1$

$b = \frac{(n-1)n}{2} - (n - 1) = \frac{1}{2}n^2 - \frac{3}{2}n - 1$

$a = \sum_{i=1}^{n-1} 1 = n - 1$

$$T(n) = 10\left[\frac{1}{2}n^2 - \frac{3}{2}n - 1\right] + 4[n - 1] + 2$$
$$T(n) = 5n^2 - 11n - 12$$

**Finding** $\boldsymbol{O(f(n))}$:

1. $\boldsymbol{f(n) = n^2}$
2. $\boldsymbol{C = 5 + 1 = 6}$
3. $\boldsymbol{|T(n_0)| \le C|f(n_0)|}$

   $\boldsymbol{\left|5n_0^2 - 11n_0 - 12\right| \le 6\left|n_0^2\right|}$

   $\boldsymbol{n_0^2 + 11n_0 + 12 \ge 0}$

   $\boldsymbol{n_0 = \left\lceil \frac{-11 \pm \sqrt{11^2 - 4 \cdot 12}}{2} \right\rceil = -1}$

$\boldsymbol{T(n)}$ **is** $\boldsymbol{O(n^2)}$ **because we found witnesses** $\boldsymbol{C = 6}$ **and** $\boldsymbol{n_0 = -1}$ **that make** $\boldsymbol{|T(n_0)| \le Cf(n_0)}$ **for** $\boldsymbol{n \ge n_0}$