Lecture 1

Dr. Frid

Topics we will cover, Background you should Have.

knowledge you should Have

- Minductive proofs ✓
- BAsic code Analysis ✓
- DeSinition of Big-O, ummitemum ✓
- Studied disserent algorithms that achieve same goal sorting - quick sort, mage sort, selection, RADIX
- DATA structure
  - Lists, dumes, Stacks, Heaps.
  - Disjoint-set
  - unionfind
- graph Atso
  - BFS, DFS, MST
  - shorter path Dijskra

---

part 1 Analysis Learn

① granth function Ω, O, Θ

② grant function for Recursive equations/ Algorithms

Part 2 Algorithm - solution choice

③ Divie and conaur. Algorithms

④ Greedy Algorithm

⑤ Dynamic Programming Algorithm

Part 3 graph Algorithm

- BFS, DFS, MST, shortest path. Bellman Dijstra's, Apsp

Part 4

Defining P, NP, NR hard, NP-complete problems

$P \subseteq NP$

ex1

T is a function of runtime

foo(int n=9)

O(1)

```
for(i=1 to i=20; i++){
    print hi
}
```

1 + 1 + 1 ≈ 20

T(n) = Runtime an input size n

↑ T(n) = O(1) T(n) = 40 = O(1)

ex2

```
foo(int n){
    c=n;
for(i=1 to i=n; i++){
    print hi;
}
```

n=10

T(n) = 2n = O(n)

$n=10$

$T(n) = \frac{n}{2}$

$O(n)$

```
for (i=1 ; i ≤ n ; i = i+2) {
    print "hi"
}
```

```
i = 1 ;
while (i ≤ n) {
    i = i * 2 ;
    print hi;
}
```

$O(\log_2 n)$

```
i = 1
while (i ≤ n)
    i = i + 1
    print hi
```

$O(n)$

$\log_2 n$

$\log_2 n+1$

1 2 3 4 5 ... 16

$n=8$

16

1, 2, 4, 8, 16

hi hi hi hi

1, 2, 4, 8, ... $2^i = n$

↑ ↑ ↑ ↑ ↑

0 1 2 3 ith iteration

$\log_2 2^i = n$ $\quad n=32$

$i = \log_2 n$

```
for (i=1 to n)
    for (j=1 to n)
        print hi
```

$O(n)$ $O(n)$ $\quad O(n^2)$

```
for (i=1 to n) { print "hello"
    if (i=7) {
        for (j=1 to n) {
            print hi
        }
    }
}
```

$n = 10$

$i = 1$ hello

$i = 2$ "hello"

$i = 3$

$i=7$ hello → hi hi

8 hello

9 hello

10 hello

$n \cdot hello + n \cdot hi \Rightarrow \simeq 2n$

$n + n^2$

ECS 122A Growth function

Definition of Algorithm

step1: We'll define computation problem

↳ you known exactly what the input is

↳ you known exactly what you want the output to be.

step2: well define procedure that transforms the input into the output

input: set limits size n.

output: A permutation of the set S such that

$S = \{a_1, a_2 \cdots, a_n\}$

$a_1 \le a_2 \le a_3 \le \cdots \le a_n$

What problem? sorting Algorithms? merge sort

---

① Does halt?

② is it correct?

③ How much memory does it use?

④ is it fast?

⑤ How does data communicate

$O(n) \Rightarrow n$

$O(n) \Rightarrow 1000^{10c}_{n}$

2^{32}

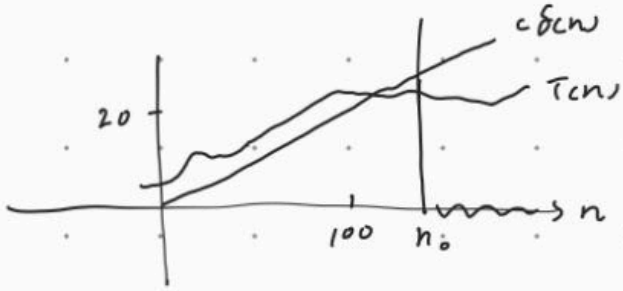
4, 3, 7 ⟶ 3, 4, 7

[7, 8, 5, 6, 8, 9, 10] $n \log n$

Let $T(n)$ = runtime of the algorithm give the size of the input is n.

$T(n)$ is $O(f(n))$ if $\exists\ c, n_0$ such that $c > 0$ $n_0 \ge 0$ AND

$\forall\ n \ge n_0\ \ T(n) \le c \cdot f(n)$



prove $T(n)=\frac{1}{2}n$ $T(n)=O(n)$

$\frac{1}{2}n \le c\cdot n$ $\exists$ $c=1/2$, $n_0=0$

$\frac{1}{2}n \le \frac{1}{2}n$

By def Big-O $T(n)=\frac{1}{2}n=O(n)$

$T(n)=5n+7$ $T(n)=O(n)$

$5n+7 \le cn$ $n_0=1$

$\le \le \le$

$5n+7n \le 12n$ $\exists$ $n_0=1$, $c=12$

prove $5n^2+8n+15$ prove $T(n)=n^2$

$5n^2+8n+15 \le cn^2$

$\le \le + \le$ $n\ge 1$

$5n^2+8n^2+15n^2 \le 28n^2$ $n_0=1$

$c=28$

$\forall_{n\ge 1}\quad 5n^2+8n+15 \le 28n^2 \Rightarrow O(n^2)$

$5n\log n+7n+8 = O(n\log n)$

$5n\log n+7n+8 \le c\,n\log n$

$\le \le$

$5n\log n+7n\log n+8n\log n$ $n\ge 10$, $\log n=10$

$5n^2=O(n)$

$5n^2 \le cn$

$5n \le c$ No constant is more than ∞

$\lim_{n\to\infty} \infty \le c$

$5n^2+8n+13 = O(n^3)$

$\leq \quad \leq \quad \leq$

$5n^3+8n^3+13n^3 \leq cn^3$

$n \geq 1$

$n_0 = 1$

$26n^3 \leq 26n^3$

$\therefore c = 26$

$\forall_{n \geq n_0} \; 5n^2+8n+13 = O(n^3)$