Big ‘O’… & Partial Order

Yue

Outline

- Asymptotic notation
- Master Method
- Partial Order basic

Asymptotic Notation

Application examples:

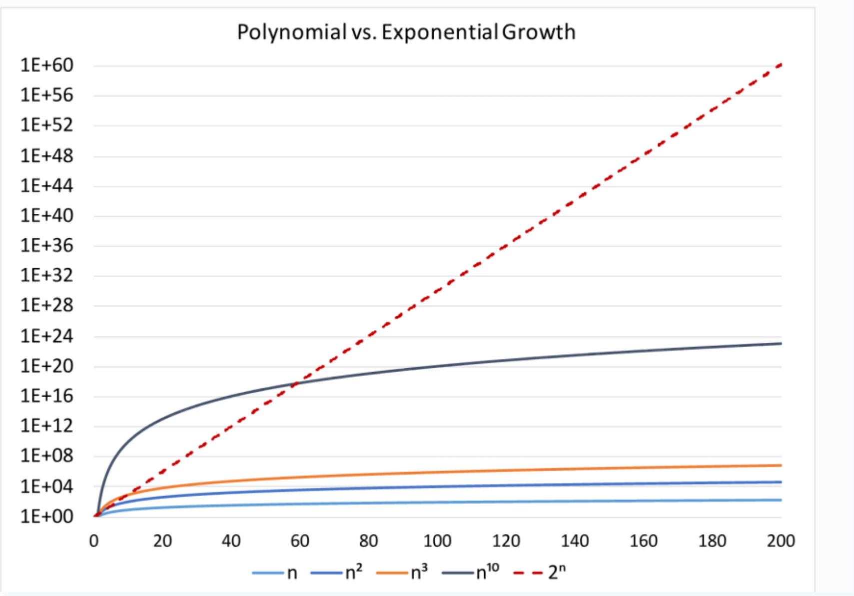
- P & NP

More formally, for a language L, we have $L \in \mathsf{P}$ if there exists a polynomial-time algorithm D such that:

- if $x \in L$, then D accepts x
- if $x \notin L$, then D rejects x

Similarly, we have $L \in \mathsf{NP}$ if there exists a polynomial-time algorithm V such that:

- if $x \in L$, then $V(x, c)$ accepts for at least one certificate c
- if $x \notin L$, then $V(x, c)$ rejects for all certificates c



- Sorting algorithm

Sort	Best	Average	Worst	Memory
Bubble	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Selection	Ω(n^2)	Θ(n^2)	O(n^2)	O(1)
Insertion	Ω(n)	Θ(n^2)	O(n^2)	O(1)
Heap	Ω(n log n) (distinct keys)	Θ(n log n)	O(n log n)	O(1)
Merge	Ω(n log n)	Θ(n log n)	O(n log n)	O(n)
Quick	Ω(n log n)	Θ(n log n)	O(n^2)	O(log n)

Definition

	Notation	Formal definition	Limit definition
Asymptotic upper bound	$f(n) = O(g(n))$	exist positive constants c and n_0 such that $0 \le f(n) \le cg(n)$ for all $n \ge n_0$	$\lim_{n\to\infty} sup\left(\frac{f(n)}{g(n)}\right) < \infty$
Asymptotic lower bound	$f(n) = \Omega(g(n))$	exist positive constants c and n_0 such that $0 \le cg(n) \le f(n)$ for all $n \ge n_0$	$\lim_{n\to\infty} inf\left(\frac{f(n)}{g(n)}\right) > 0$
Asymptotic tight bound	$f(n) = \Theta(g(n))$	exist positive constants c1, c2, and n_0 such that $0 \le c1g(n) \le f(n) \le c2g(n)$ for all $n \ge n_0$	The two above

Stirling approximation: $n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$

What's the time complexity of the following algorithm?

```c
void insertionSort(int arr[], int n)
{
    int i, key, j;
    for (i = 1; i < n; i++) {
        key = arr[i];
        j = i - 1;
        while (j >= 0 && arr[j] > key) {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}
```

```c
int partition(int arr[], int low, int high)
{
    int pivot = arr[high];
    int i = (low - 1);

    for (int j = low; j <= high - 1; j++) {
    if (arr[j] < pivot) {
            i++;
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}

void quickSort(int arr[], int low, int high)
{
    if (low < high) {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}
```

Master Theorem

What's the complexity of merge sort?

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n)$$

Algorithm $MergeSort(A[1..n]$: array of n integers$)$:
  If $n = 1$ **return** A
  $m := \lfloor n/2 \rfloor$
  $L := MergeSort(A[1..m])$
  $R := MergeSort(A[m+1..n])$
  Return $merge(L, R)$

Subroutine $merge(A[1..m], B[1..n])$:
  If $m = 0$ **return** B
  If $n = 0$ **return** A
  If $A[1] > B[1]$ **return** $B[1] + merge(A[1..m], B[2..n])$
  Return $A[1] + merge(A[2..m], B[1..n])$

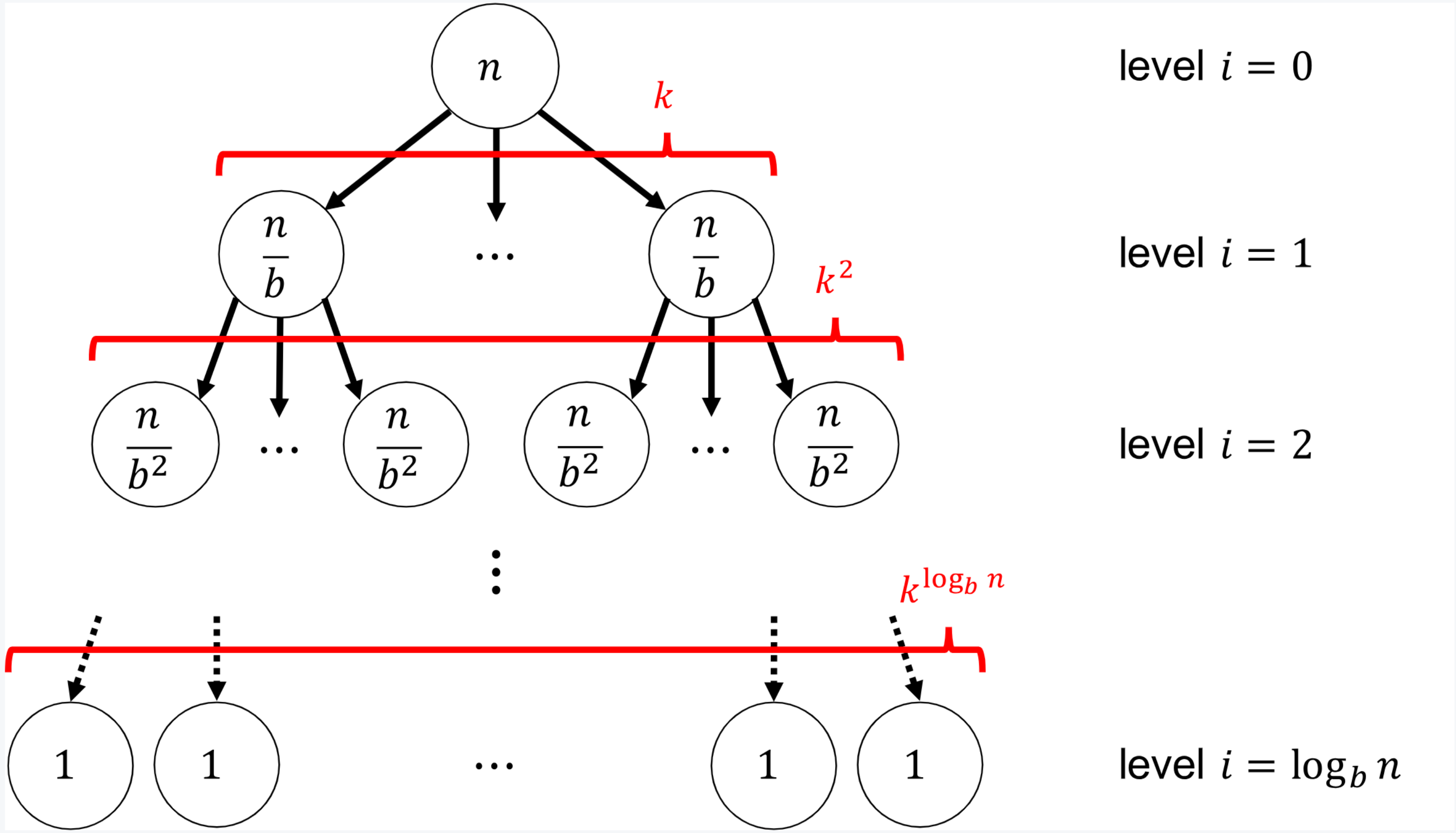
Special case:

$$T(n) = kT\left(\frac{n}{b}\right) + \mathrm{O}(n^d \log^w n)$$
$$T(n) = kT\left(\left\lfloor\frac{n}{b}\right\rfloor\right) + \mathrm{O}(n^d \log^w n)$$
$$T(n) = kT\left(\left\lceil\frac{n}{b}\right\rceil\right) + \mathrm{O}(n^d \log^w n)$$

$$T(n) = \begin{cases} \mathrm{O}(n^d \log^w n) & \text{if } k/b^d < 1 \\ \mathrm{O}(n^d \log^{w+1} n) & \text{if } k/b^d = 1 \\ \mathrm{O}(n^{\log_b k}) & \text{if } k/b^d > 1 \end{cases}$$

If $T(n) = aT(n/b) + f(n)$ *(for constants* $a \geq 1$, $b > 1$*), then*

1. $T(n) = \Theta(n^{\log_b a})$ *if* $f(n) = O(n^{\log_b a - \varepsilon})$ *for some constant* $\varepsilon > 0$.
2. $T(n) = \Theta(n^{\log_b a} \lg n)$ *if* $f(n) = \Theta(n^{\log_b a})$.
3. $T(n) = \Theta(f(n))$, *if* $f(n) = \Omega(n^{\log_b a + \varepsilon})$ *for some constant* $\varepsilon > 0$, *and if* $af(n/b) \leq cf(n)$ *for some constant* $c < 1$ *and all sufficiently large* n *(regularity condition).*



$$T(n) = kT\left(\frac{n}{b}\right) + \mathrm{O}(n^d)$$
$$T(n) = kT\left(\left\lfloor\frac{n}{b}\right\rfloor\right) + \mathrm{O}(n^d)$$
$$T(n) = kT\left(\left\lceil\frac{n}{b}\right\rceil\right) + \mathrm{O}(n^d)$$

$$T(n) = \begin{cases} \mathrm{O}(n^d) & \text{if } k/b^d < 1 \\ \mathrm{O}(n^d \log n) & \text{if } k/b^d = 1 \\ \mathrm{O}(n^{\log_b k}) & \text{if } k/b^d > 1 \end{cases}$$

Divide and conquer
“分而治之”

there are $1 + \log_b n$ total levels in the recursion.

How much work is done in each subproblem: $\mathrm{O}\left(\left(\frac{n}{b^i}\right)^d\right) = \mathrm{O}\left(\frac{n^d}{b^{id}}\right) = b^{-id} \cdot \mathrm{O}(n^d)$

With k^i subproblems at level i, the total work T_i at level i is $T_i = \frac{k^i}{b^{id}} \cdot \mathrm{O}(n^d) = \left(\frac{k}{b^d}\right)^i \cdot \mathrm{O}(n^d)$

$$T = \sum_{i=0}^{\log_b n} T_i$$

Examples

$$T(n) = 3T(n/2) + n^2$$

$$T(n) = 4T(n/2) + n^2$$

$$T(n) = T(n/2) + 2^n$$

$$T(n) = 16T(n/4) + n$$

Answer:

$$T(n) = 3T(n/2) + n^2 \implies T(n) = \Theta(n^2)$$

$$T(n) = 4T(n/2) + n^2 \implies T(n) = \Theta(n^2 \log n)$$

$$T(n) = T(n/2) + 2^n \implies \Theta(2^n)$$

$$T(n) = 16T(n/4) + n \implies T(n) = \Theta(n^2)$$

Partial Order

Poset (P, ≤)

- Reflexive: $\forall x \in P, x \leq x$
- Antisymmetric: $\forall x, y \in P, x \leq y \wedge y \leq x \rightarrow x = y$
- Transitive: $\forall x, y, z \in P, x \leq y \wedge y \leq z \rightarrow x \leq z$

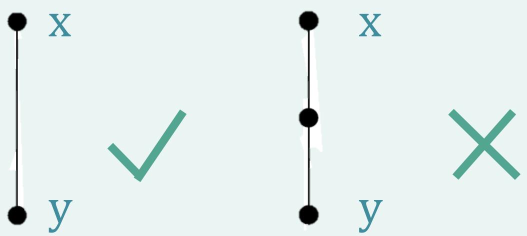
(maybe for some x, y no relation between them)

e.g.: (all subset of a set X, ⊂);
A directed graph without cycle

+ dichotomy ∀x, y ∈ P (x ≤ y or y ≤ x)   and if original order relation kept

=> Linear order   linear extention

y cover x



Minimal/maximal: no larger/smaller element
(may not unique)

Comparable with every element

Minimum/maximum(unique if exist)

Example

We naturally order the numbers in $A_m=\{1,2,\ldots,m\}$ with "less than or equal to," which is a partial ordering. We define an ordering, $\preceq$ on the elements of $A_m\times A_n$ by

$$(a,b)\preceq(a',b') \iff a\leq a' \text{ and } b\leq b'$$

1. Prove that $\preceq$ is a partial ordering on $A_m\times A_n$.
2. Draw the Hasse diagrams for $\preceq$ on $A_2\times A_2$, $A_2\times A_3$
3. What is the minimal element? What is the minimum element?

Set $L=\{3,5,7,15,35,45,105\}$

1. $(L, |$(divisibility)) is a poset
2. What is the minimal element? What is the minimum element?

End

~~QAQ~~Q&A

Reference

- Umich EECS376 Notes