CS F364: Design & Analysis of Algorithm

01 Introduction to Algorithm

Dr. Kamlesh Tiwari
Assistant Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Jan 18, 2021 ONLINE (Campus @ BITS-Pilani Jan-May 2021)

`http://ktiwari.in/algo`

Logistics: (CS F364) Design & Analysis of Algorithms

- M W F (3:00PM-3:50PM) online
  http://meet.google.com/jto-vjtw-bsd
- Jointly to be taught by
  Dr. Abhishek Mishra (IC) and **Dr. Kamlesh Tiwari**.
- Grading
  - Tutorial Quiz (32%) 4 of 8% each, Open Book
  - Mid Semester Exam (28%) Open Book
  - Comprehensive Exam (40%) Open Book

Learn algorithm design techniques like Divide and Conquer, Greedy, Dynamic Programming, Approximation Algorithms, and Randomized Algorithms. Explore topics like Computational Complexity *etc.*

- **Books:**

[1] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, 3rd Edition, PHI, 2012.
[2] S. Arora, B. Barak, Computational *Complexity: A Modern Approach*, Cambridge University Press, 2009
[3] J.Kleinberg, E. Tardos, *Algorithm Design*, Pearson, 2013.

Introduction

- Computational Problems
- Algorithms: input, output, definiteness, finiteness, effectiveness
- `Pseudo code`
- Input size
- Analysis
  - **Kind of resources**[1]: time, space, number of gates ...
  - **Cases**: Best, Worst and Average
- Correctness: initialize well, maintain invariance and terminate
- Order of growth: O, o, θ, ω, Ω zoo
- Insertion and Merge sort

[1] Complexity is a function

Insertion Sort

Incremental algorithm paradigm:

Algorithm 1: INSERTION-SORT (A)

```
1 for j = 2 to A.length do
2     key = A[j]
3     i = j − 1
4     while i > 0 and A[i] > key do
5         A[i + 1] = A[i]
6         i = i − 1
7     A[i + 1] = key
```


Analyse Insertion Sort

INSERTION-SORT (A)	cost	times
1 **for** $j = 2$ **to** $A.length$	c_1	n
2 $key = A[j]$	c_2	$n-1$
3 $i = j - 1$	c_3	$n-1$
4 **while** $i > 0$ **and** $A[i] > key$	c_4	$\sum_{j=2}^{n} t_j$
5 $A[i+1] = A[i]$	c_5	$\sum_{j=2}^{n}(t_j - 1)$
6 $i = i - 1$	c_6	$\sum_{j=2}^{n}(t_j - 1)$
7 $A[i+1] = key$	c_7	$n-1$

- Best case $T(n) = O(n)$
- Worst case $T(n) = O(n^2)$
- Average ?

Merge sort

Divide and conquer paradigm: Divide, Conquer and Combine

$$T(n) = \begin{cases} \Theta(1) & \text{if } n \le c \\ aT(n/b) + D(n) + C(n) & \text{otherwise} \end{cases}$$

MERGE-SORT (A,p,r)
1 **if** $p < r$
2 $q = \lfloor (p+r)/2 \rfloor$
3 MERGE-SORT (A,p,q)
4 MERGE-SORT (A,q+1,r)
5 MERGE (A,p,q,r)

```
MERGE(A, p, q, r)
1  n1 = q − p + 1
2  n2 = r − q
3  let L[1..n1 + 1] and R[1..n2 + 1]
   be new arrays
4  for i = 1 to n1
5      L[i] = A[p + i − 1]
6  for j = 1 to n2
7      R[j] = A[q + j]
8  L[n1 + 1] = ∞
9  R[n2 + 1] = ∞
10 i = 1
11 j = 1
12 for k = p to r
13     if L[i] ≤ R[j]
14         A[k] = L[i]
15         i = i + 1
16     else A[k] = R[j]
17         j = j + 1
```

Average case $T(n) = O(n \log n)$. Best and Worst?

Thank You!

Thank you very much for your attention! (Reference[2])

Queries ?

[2] [1] Book - *Introduction to Algorithm*, By THOMAS H. CORMEN, CHARLES E. LEISERSON, RONALD L. RIVEST, CLIFFORD STEIN