# CS6100 – Topics in Design and Analysis of Algorithms

## Beyond Worst Case Analysis

# CS6100 – Topics in Design and Analysis of Algorithms

## Beyond Worst Case Analysis

Instructor:

Raghavendra Rao B V (BSB 354, bvrr@cse.iitm.ac.in, rkunjathaya@gmail.com)

# CS6100 – Topics in Design and Analysis of Algorithms

## Beyond Worst Case Analysis


Instructor:

Raghavendra Rao B V (BSB 354, bvrr@cse.iitm.ac.in, rkunjathaya@gmail.com)

TA for the course:

| | |
|---|---|
| CS18D015@smail.iitm.ac.in | SUGYANI MAHAPATRA |

# Course links

- Google form for providing contact details:
  `https://forms.gle/4NEKRMHbN2w9oF636`

# Course links

- Google form for providing contact details:
  `https://forms.gle/4NEKRMHbN2w9oF636`
  Everyone should fill this form. The email provided will be used for adding to the google group. Contact no will be used only if necessary.

# Course links

- Google form for providing contact details:
  `https://forms.gle/4NEKRMHbN2w9oF636`
  Everyone should fill this form. The email provided will be used for adding to the google group. Contact no will be used only if necessary.
- Google group for the course:
  `2021CS6100@googlegroups.com`

# Course links

- Google form for providing contact details:
  `https://forms.gle/4NEKRMHbN2w9oF636`
  Everyone should fill this form. The email provided will be used for adding to the google group. Contact no will be used only if necessary.
- Google group for the course:
  `2021CS6100@googlegroups.com`
  Everyone who fill the above mentioned form by 03 Feb 5pm, will be added to this group.
- Moodle for the course: `https://courses.iitm.ac.in` (course no 6100)

# Course links

- Google form for providing contact details:
  `https://forms.gle/4NEKRMHbN2w9oF636`
  Everyone should fill this form. The email provided will be used for adding to the google group. Contact no will be used only if necessary.
- Google group for the course:
  `2021CS6100@googlegroups.com`
  Everyone who fill the above mentioned form by 03 Feb 5pm, will be added to this group.
- Moodle for the course: `https://courses.iitm.ac.in` (course no 6100)
- Google drive for lectures: `tinyurl.com/3fwoyy7t`

# Course links

- Google form for providing contact details:
  `https://forms.gle/4NEKRMHbN2w9oF636`
  Everyone should fill this form. The email provided will be used for adding to the google group. Contact no will be used only if necessary.
- Google group for the course:
  `2021CS6100@googlegroups.com`
  Everyone who fill the above mentioned form by 03 Feb 5pm, will be added to this group.
- Moodle for the course: `https://courses.iitm.ac.in` (course no 6100)
- Google drive for lectures: `tinyurl.com/3fwoyy7t`
- Course Web page: `https://theory.cse.iitm.ac.in`

# Course links

- Google form for providing contact details:
  `https://forms.gle/4NEKRMHbN2w9oF636`
  Everyone should fill this form. The email provided will be used for adding to the google group. Contact no will be used only if necessary.
- Google group for the course:
  `2021CS6100@googlegroups.com`
  Everyone who fill the above mentioned form by 03 Feb 5pm, will be added to this group.
- Moodle for the course: `https://courses.iitm.ac.in` (course no 6100)
- Google drive for lectures: `tinyurl.com/3fwoyy7t`
- Course Web page: `https://theory.cse.iitm.ac.in`

# Course Objectives

- To explore paradigms beyond worst-case analysis of algorithms such as smoothed and probabilistic analysis.
- To understand various techniques involved in smoothed and probabilistic analysis.
- Expose the student to the state of art in algorithm analysis techniques.

# Course Content

- Basics of probability: Distributions: Gaussian, exponential etc. Markov's inequality, Chebyshev's, Chernoff's, Martingales and Talagrand's isoperimetry (Roughly 1-2 Weeks)
- Smoothed analysis of Algorithms: Introduction to Smoothed analysis. Examples of smoothed analysis for problems such as Knapsack, Integer Programming, TSP: 2-OPT. Smoothed number of pareto optimal solutions. Other local search algorithms. Partitioning algorithm. K-means clustering. Stability: algorithms for stable instances. Perturbation stable clustering. Stable instances.(6-7 weeks)
  Probabilistic Analysis of Algorithms: Probabilistic analysis of Euclidean optimization problems. Concentration bounds. Other probabilistic models. (4 weeks)

# References

**Books/Notes:**

- **Probablity Theory and Combinatorial Optimization** by Michael Steele
- **Probability Theory of Classical Euclidean Optimization Problems** by Joseph Yukich
- **Beyond Worst case analysis of Algorithms** – Lecture notes by Bodo Manthey, University of Twente, The Netherlands
- **Probabilistic Analysis of Algorithms** – Lecture noted by Heiko Röglin, University of Bonn, Germany.

# Evaluation Scheme

| Activity | Weight (%) |
|---|---|
| Evaluated Assignments (2 total) | 10% (10% × 2) |
| Mid Sem | 15% |
| End Sem | 15% |
| Project + Presentation | 50% |

# Evaluated Assignments

- Take home exams conducted through moodle.
- You will be given a 15 days time for uploading your solutions.
- Students will have to solve the problems individually and upload their solutions before the deadline.
- It is fine to submit possibly incorrect but your own solution rather than discussing/copying the solution from elsewhere.
- Grading will be done based on the innovations and efforts in the solution rather than perfectness of a solution.
- For example, an incorrect solution but involves a completely out of the box thinking can potentially fetch full marks.
- No discussion is allowed among the students and any violation will be dealt with strictly as per Institute's policy on academic dishonesty.

# Exams

- Take home exams that will conducted through moodle.
- There will be a total of two exams.
- There will be a window of 48 hours to attempt the exam. Once started, you will have to submit the answers within 3 hours.
- There will be around 3-4 questions. Questions will be randomized from a pool.
- Students will have to answer the questions individually and upload the answers before the deadline.
- No discussion is allowed among the students and any violation will be dealt with strictly as per Institute's policy on academic dishonesty.

# Dates (Tentative)

| Activity | Date |
|---|---|
| Assignment - 1 | Feb 10 - 24 |
| Assignment - 2 | March 10 - 30 |
| Assignment - 3 | optional |
| Mid Sem | Mar 12 - 14 |
| End Sem | Apr 30 - May 02 |

# Project

- The default form of the project is to do a detailed reading of a research article (to be chosen from a list) followed by presentations and a viva. Highly motivated one's can also pursue a related research question.

# Project

- The default form of the project is to do a detailed reading of a research article (to be chosen from a list) followed by presentations and a viva. Highly motivated one's can also pursue a related research question.
- Also thinking of having hybrid projects were a part will be implementing an algorithm and doing a performance evaluation followed by a theoretical explanation.
- Your thoughts about project will be important in finalizing the project component. Please fill https://forms.gle/6nB4zvGt8SZWxJfN6 by **Feb 4, 5pm**. Form is kept anonymous, so feel free to give any kind of opinion/suggestions without the fear of being judged.

# Lectures

- Four lectures per week initially. After the project commences, will be reduced to three lectures per week. All lectures will be live through Google meet. Will be recorded for convenience. please remind me to record if i forget to!!
- There will be occasional tutorials.

# Policy on academic dishonesty

- Any form of academic dishonesty will be dealt with severe punishment as per institute's policy.
- Exams and evaluated assignments should be attempted by students individually and any form of plagiarism, discussion etc are prohibited. Any violation of this will be reported to the disciplinary committee of the institute (DISCO)

# Worst case analysis

A - be an algorithm

$t(\bullet)$ - some performance measure of A

e.g.: time complexity
quality of output

worst case performance (A) = $\max_{x \text{ inputs of length } n} t(x)$

mergesort : $O(n \log n)$

quicksort $O(n^2)$

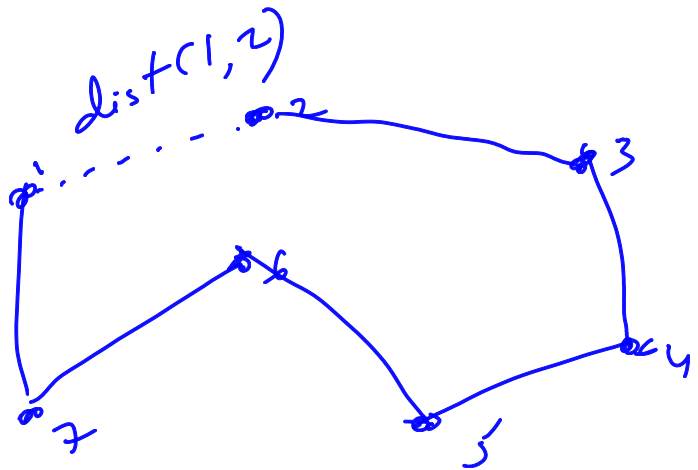
## Example 1

TSP (Traveling Salesman Problem)

i/p: n cities $p_1, \ldots, p_n$ with distance b/n every pair of cities

&

o/p: A tour is a cycle $p_{i_1}, p_{i_2}, \ldots p_{i_n}, p_{i_1}$ s.t $\sum_{j=1}^{n} dist(p_{i_j}, p_{i_{j+1 \bmod n+1}})$ is the minimum




TSP is NP hard

$2^{100}$

$n = 100$

$2^n$

$n = 30$

$2^{20}$

$2^{30}$

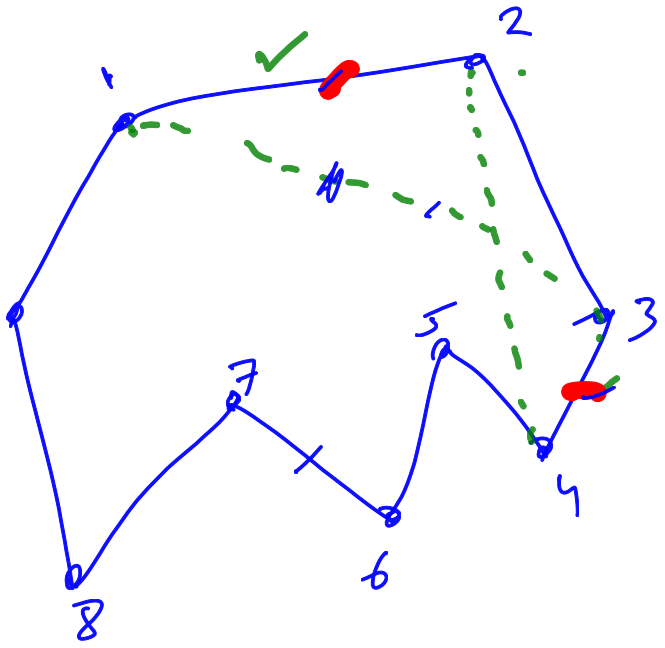
If obtaining exact solution is hard
people settle for close enough approximation

Approximation Algorithm

## 2-opt algorithm

① Let $T$ be some tour.

② For every pair
e, f of 2 edges in $T$
See if the new tour
obtained by swapping
endpoints of e & f has
smaller cost. If yes
update $T$ to that tour.




1, 3, 2, 4, 5, 6, 7, 8, 9

k-opt

(3) repeat the above until no further improvement is possible.