



BOSSCODER  
ACADEMY



# MASTERCLASS NOTES

# GRAPHS

Starts @ 0905 PM

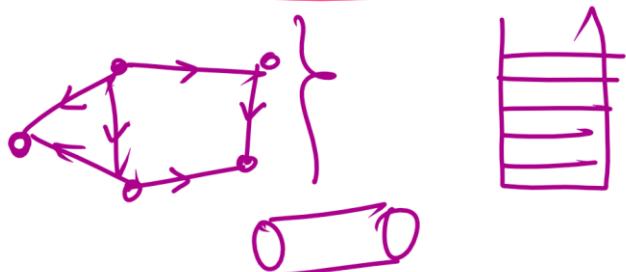
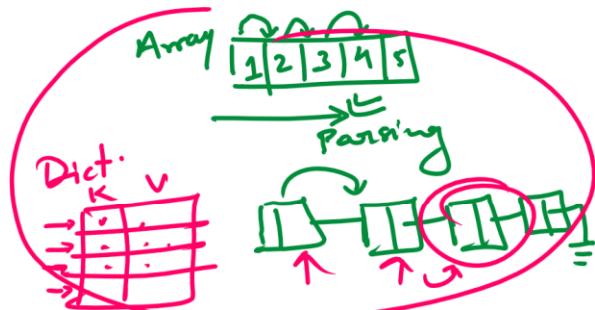
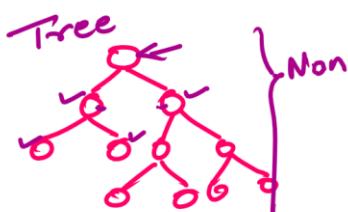
↳ By Ashish Prasad  
 10 yrs full stack  
 Msft → ↳ C#  
 SNGAMES → Slot Game  
 HXGN → 911 - Amb  
 fi.  
 Pol.

- Intro →
- Basic Graphs ✓
- Representations
- 1 Qn Amzn
- Pitch BCA
- 2 Inter

swift  
Swift UP

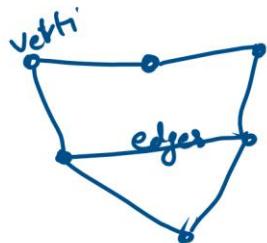
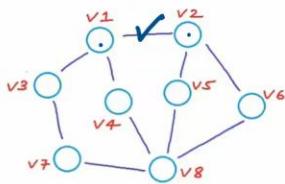
Data str:

linear ←  
 non linear



Graph "G" is an ordered pair of set "V" of vertices & a set "E" of edges

$$G = (\bar{V}, \bar{E})$$

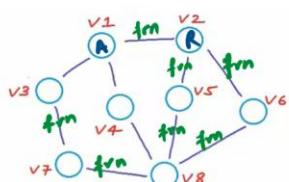


graph:  
 $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$

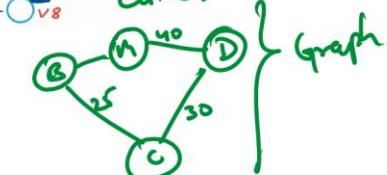
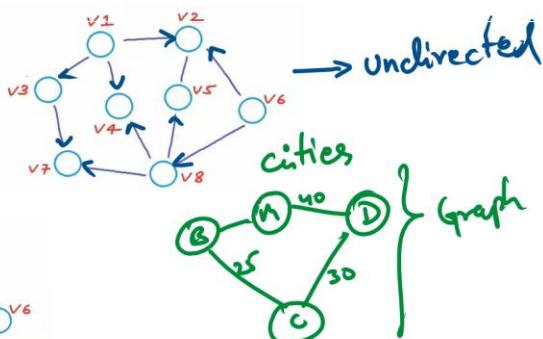
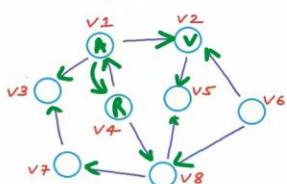
E:  $\{(v_1, v_2), (v_2, v_3), (v_2, v_5), (v_4, v_8), (v_8, v_5), (v_5, v_6), (v_6, v_7), (v_7, v_8), (v_1, v_8), (v_1, v_3)\}, \dots$

Directed & undirected

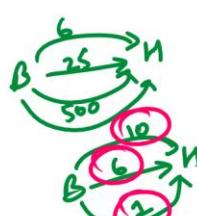
facebook



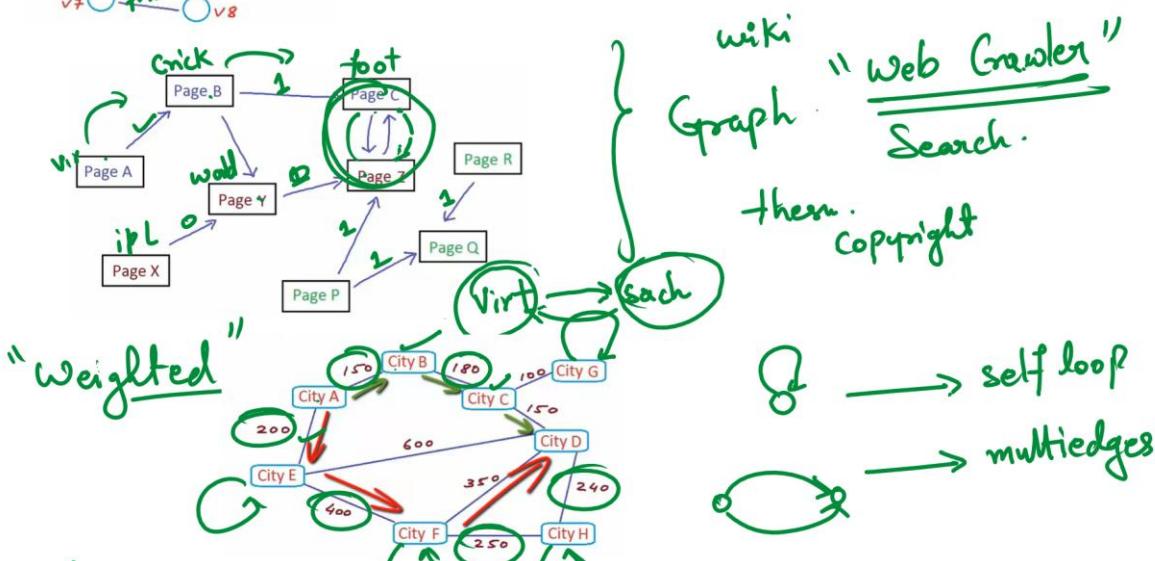
Pastagram



"Weighted"



multiedge weighted graph.



self loop

multiedges



Vertices

## Properties of Graph

Number of edges.

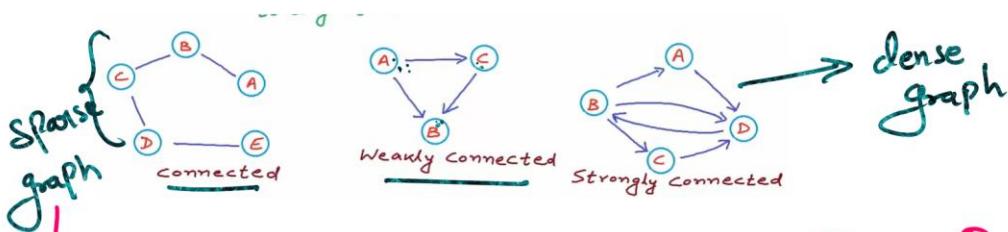
if  $|V| = n$

then

$$0 \leq E \leq n(n-1)$$

assuming we don't have (selfloop or multiedge)

$$0 \leq E \leq \frac{n(n-1)}{2}$$



no. edges are less

$\rightarrow$  cyclic Graph

directed acyclic graph } DAG } Topological Sort.

## Graph Representation

$$G = (V, E)$$

$V = \{v_1, v_2, \dots\}$  int array

$$E = \{(v_1, v_2), (v_3, v_4), \dots\}$$

class Edge { List<Edge> edges = new List<Edge>();

{ string startV  
string endV  
int weight }



T.C. - find adjacent node to  $v_1 \Rightarrow v_2 \& v_5$

$\alpha |E|$

check if given nodes are connected  $\alpha |E|$

Crude way

SV EV

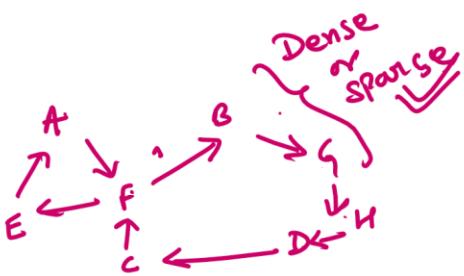
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

obj Array

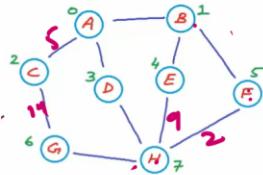
## Matrix

int [ ] mat + new mat [8][8]

	A	B	C	D	E	F	G	H
A	0	1	1	1	0	0	0	0
B	1	0	0	0	1	0	0	0
C	2	1	0	0	0	0	1	0
D	3	1	0	0	0	0	0	1
E	4	0	1	0	0	0	0	1
F	5	0	1	0	0	0	0	1
G	6	0	0	1	0	0	0	1
H	7	0	0	0	1	1	2	1



# GRAPHS



undirected  
graph.

Symmetric along diagonal

All adjacent node to 'e' = AEF

A	B	C	D	E	F	G	H	
A	0 0 0 0 0 1 1 0	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
B	0 0 0 0 0 0 1 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
C	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
D	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
E	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
F	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0
G	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

matrix  
is good for  
Dense  
graph  
but not for  
sparse

Starts @ 09:05 PM

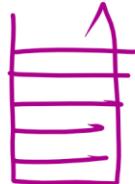
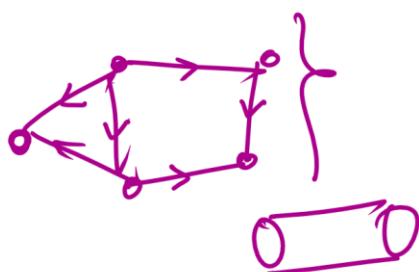
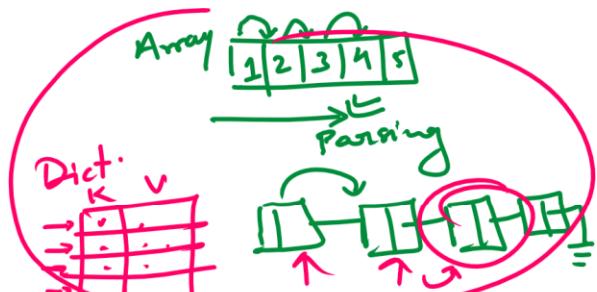
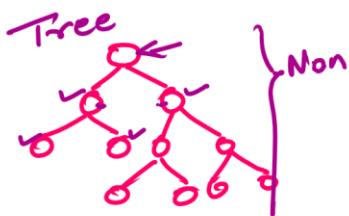
↳ By Ashish Prajapati  
10 yrs full stack  
Msft → C#  
SNGAMES → React  
HXGN → Slot Game  
911 - Amb.  
fi.  
Pol.

- Intro →
- Basic Graphs ✓
- Representations
- 1 Qn Amzn
- Pitch BCA
- 2 Inter

swift  
Swift UP

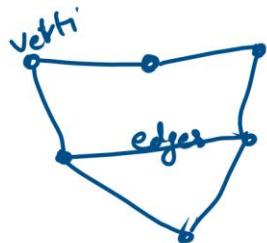
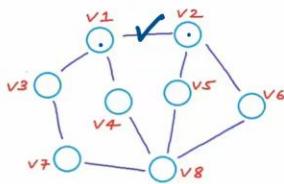
Data Str:

linear ←  
non linear



Graph "G" is an ordered pair of set "V" of vertices & a set "E" of edges

$$G = (\bar{V}, \bar{E})$$

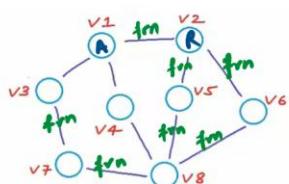


graph  
 $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$

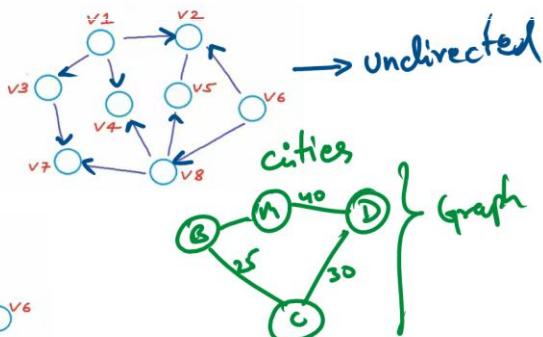
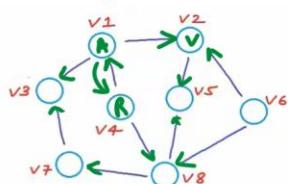
$E = \{(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_5), (v_5, v_6), (v_6, v_7), (v_7, v_8), (v_8, v_1)\}, \dots\}$

Directed & undirected

facebook

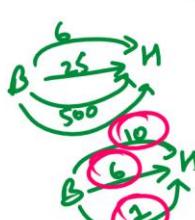


Pastagram

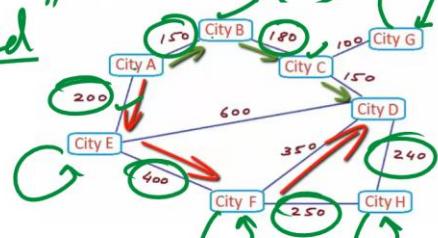


wiki  
 Graph "Web Crawler"  
 Search.  
 them.  
 copyright

"Weighted"



} multiedge weighted graph.



self loop

} multiedges



## Properties of Graph

Number of edges.

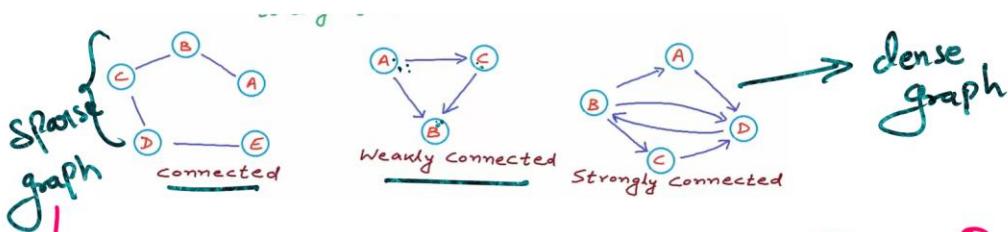
if  $|V| = n$

then

$$0 \leq E \leq n(n-1)$$

assuming we don't have (selfloop or multiedge)

$$0 \leq E \leq \frac{n(n-1)}{2}$$



no. edges are less

$\rightarrow$  cyclic Graph

directed acyclic graph } DAG } Topological Sort.

## Graph Representation

$$G = (V, E)$$

$V = \{v_1, v_2, \dots\}$  int array

$$E = \{(v_1, v_2), (v_3, v_4), \dots\}$$

congrowing  
no limit  
 $v_i$

class Edge { List<Edge> edges = new List<Edge>()

{ string startV  
string endV  
int weight }



T.C. - find adjacent node to  $v_1 \Rightarrow v_2 \& v_5$

$\alpha |E|$

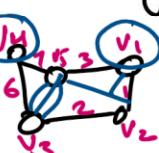
check if given nodes are connected  $\alpha |E|$

SV EV

1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

obj Array

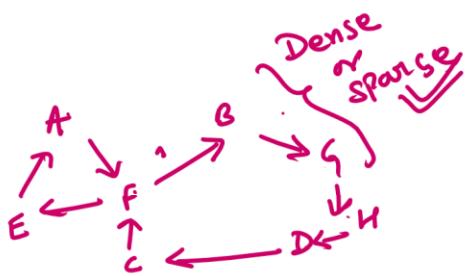
Crude way



## Matrix

int [ ] mat + new mat [8] [8]

	A	B	C	D	E	F	G	H
A	0	1	1	1	0	0	0	0
B	1	0	1	0	0	1	0	0
C	2	1	0	0	0	0	1	0
D	3	1	0	0	0	0	0	1
E	4	0	1	0	0	0	0	1
F	5	0	1	0	0	0	0	1
G	6	0	0	1	0	0	0	1
H	7	0	0	0	1	1	2	1

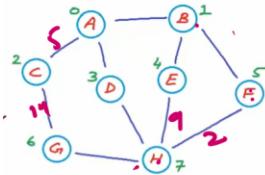


Symmetric along diagonal

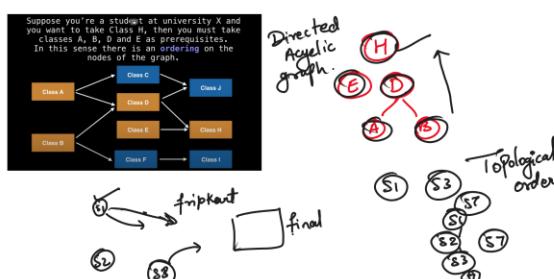
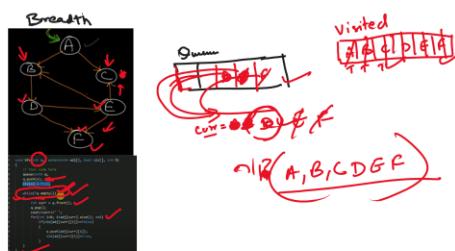
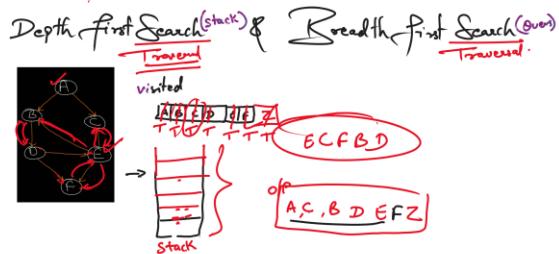
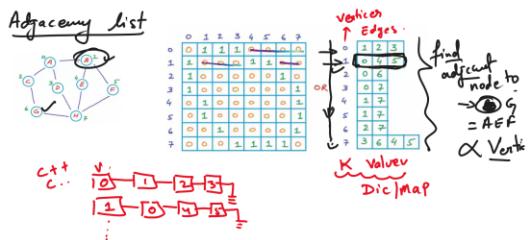
All Adjacent node to 'e' = A-E

A	B	C	D	E	F	G	H
A	0	0	0	0	0	1	0
B	0	0	0	0	0	0	1
C	0	0	0	0	0	0	0
D	0	0	1	0	0	0	0
E	0	0	0	0	0	0	1
F	0	1	0	0	0	0	0
G	0	0	0	0	0	0	1

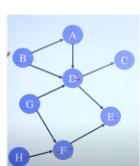
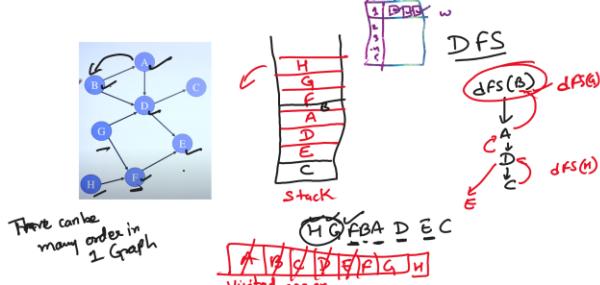
matrix  
is good for  
Dense  
graph  
but not for  
sparse



undirected  
Graph.



BADC ✓  
BDCA ✓



# ABOUT BOSSCODER

Bosscoder is an online upskilling platform for techies. We help learners upskill in tech roles to get them placed at top tech companies. We do so through our structured & mentored program designed by industry experts.

## USP of our program include:

### ✨ **STRUCTURED CURRICULUM:**

Covers everything you need to get placed at top tech companies: Problem solving in DS & Algo, CS Fundamentals, System Design (HLD + LLD), Full Stacks & Queues Projects

### ✨ **LIVE CLASSES:**

An active learning classroom program taught by engineers working at companies like Microsoft, PayPal, Amazon

### ✨ **1:1 MENTORSHIP & MOCK INTERVIEWS:**

Personal mentors from top tech companies help you provide the right guidance, feedback, and support.

### ✨ **24/7 DOUBT SUPPORT:**

Through our army of Teaching Assistants

### ✨ **INDUSTRY-RELEVANT PROJECTS:**

Full Stacks & Queues specialization with Industry-relevant projects

### ✨ **PLACEMENT SUPPORT:**

Providing opportunities to tech engineers in eminent startups & top tech companies.

# BUILD YOUR CAREER WITH US

-  **400+** Alumni placed at Top Product-based companies.
-  Highest package of **86 LPA**
-  Average package of **22 LPA**.
-  Resume reviewed and interview scheduled for **1000+** students



**Lakshmi susmitha**  
Software Engineer II, JP Morgan

## Service Based to JP Morgan in 4 months

From a tier-3 college to working in service-based companies, my thirst to join a product-based company didn't go away. BossCoder helped me provide a very detailed path from coding to system design. The way of teaching, and 1:1 mentorship helped me a lot.

Before  
IBM  
Application Engineer



After  
JP Morgan  
Software Engineer II



**Dheeraj Barik**  
Software Engineer 2, Amazon

## System Engineer at Service Based to SDE 2 at Amazon

Working in Infosys, I was looking for a platform to prepare for interviews of product-based companies. I found BossCoder has a highly structured program covering DSA, System Design etc. in detail. Top-quality instructors and mock interviews proved helpful for me.

Before  
Infosys  
Systems Engineer



After  
Amazon  
SDE 2



**Vishal Srivastava**  
Software Developer, Barclays

### Service Based to London Based Bank

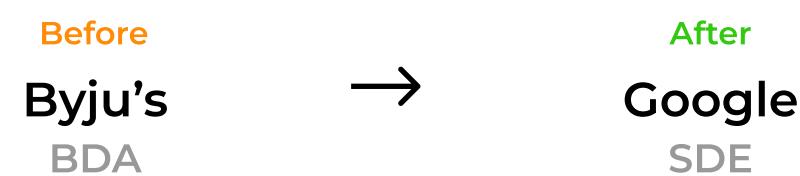
Doing self-prep, I couldn't even pass online assessments of Top companies. But the change BossCoder Academy brought into my preparation is phenomenal. Crucial topics taught in Live classes like DSA, HLD, and LLD, and my personal mentor's guidance ensured I clear my dream company.



**Ujesh Nada**  
Software Development Engineer, Google

### Business Development Associate to SDE at Google

I self-prepared DSA for 8 months, without any results. I joined BossCoder Academy since I wanted to be mentored by industry experts, and it proved to be a great decision for my DSA and System Design preparation. Their 1-on-1 mentor sessions and Live classes helped me transform my career.



**Rakesh Kumar Satapathy**  
Sr. Developer, Hashedin

### Bsc. Graduate stuck in service based to Hashedin

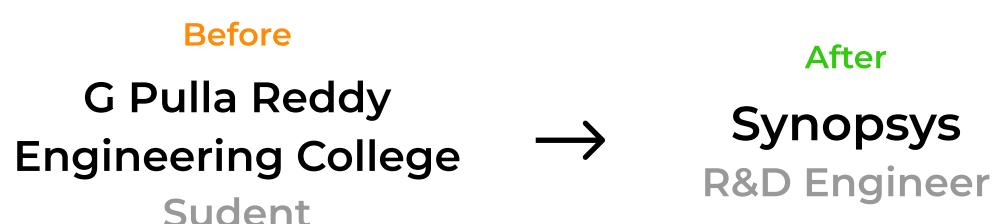
Stuck in a service-based company with no exposure, I always believed that I can realize my dream life, but didn't know how. BossCoder Academy showed the right path to coding geek inside me. Their world-class curriculum and personal mentorship enabled me to switch to my dream role.



**Harshith Ravinoothala**  
R&D Engineer 1, Synopsys

### Tier 3 College Student to Product Based Company

I always wanted to get into a product based company, but being a tier 3 college student lacked exposure to coding. BossCoder Academy helped me gain confidence in DSA and core subjects like OS, DBMS and System Design. Instant Mentor support helped me stay clear of doubts.

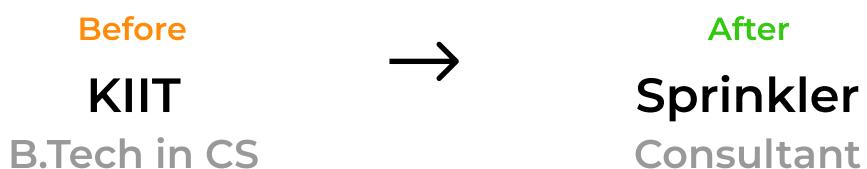




**Sarveshwar Neogi**  
Consultant, Sprinkler

### Clueless college student to Consultant at Sprinkler

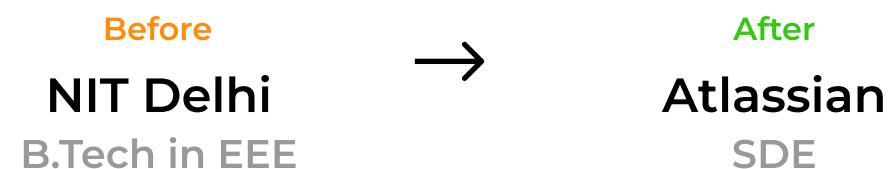
I was wasting my time in college, with no idea how to improve my coding skills. The structured roadmap provided by BossCoder transformed me into a Tech Rockstar. In-depth live lectures and daily handpicked questions helped me become consistent in problem-solving.



**Aarushi Jain**  
Software Development Engineer, Atlassian

### No interest in coding to SDE at Atlassian

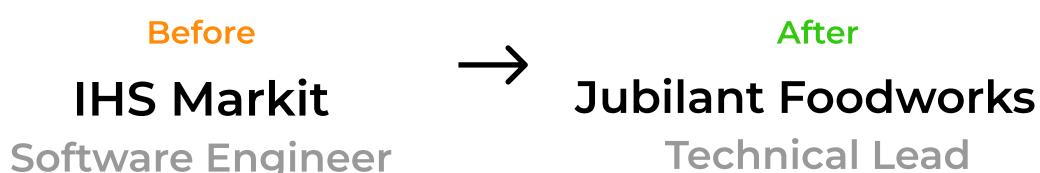
My journey in college was a roller coaster ride, and I wasted lots of time in learning from free resources. BossCoder Academy's excellent curriculum helped me become consistent and invest my efforts in the right direction. With my mentor's guidance, I received offers from Amazon and Atlassian.



**Sumedha Khandelwal**  
Technical Lead at Jubilant Foodworks

### Scared of Technical Interviews to Technical Lead

Having 7 years of experience, I always believed there is more to achieve in my career but failed in technical interviews. BossCoder helped me gain confidence to face technical interviews and I received many offers. Mock interviews and mentor feedback helped a lot.



**Irshad K**  
Software Engineer, ByteDance

### NIT Delhi to SDE in Singapore

I am among those talented students who require proper guidance to prosper. Cracking ByteDance was possible due to the guidance of my personal mentor at BossCoder Academy. Their structured curriculum helped me gain confidence in DSA and System Design.





# Want to Upskill?

Go through our website, or email us at  
[ask@bosscoderacademy.com](mailto:ask@bosscoderacademy.com)

**VISIT WEBSITE**  
[www.bosscoderacademy.com](http://www.bosscoderacademy.com)

A screenshot of a mobile application for BossCoder Academy. The left side shows a dark sidebar with navigation options: Home, Payment, My Course (selected), My mentor, Placement, Calender, Store, Problems, Lead board, Refer Earn, Profile, and Support. The main content area is titled 'My Courses' and displays four course modules: 'Beginners Lectures' (100% completed, 20/20 problems solved), 'Advance DSA' (80% completed, 15/20 problems solved), 'High Level design' (0% completed, 00/20 problems solved), and 'Low level design' (0% completed, 00/20 problems solved). Each module has a 'View Module' button. At the bottom of the main screen, it says 'Advance DSA' with 'Full course context' and 'Full report' buttons. On the right side, there's a profile section for 'R. Ekunde' with a welcome message, a notification bell, and a circular progress bar. Below the profile are performance metrics: Points (1856 / 15/1000), Streak (32 days), Solution (64 / 100), and Rating (4.8 / 5).