📘 Compiler Design UNIT–V PYQ Analysis

🔥 Code Optimization PYQ Analysis (RGPV)

Maine uploaded PYQ papers analyze kiye aur UNIT–V se related sabhi repeated aur important questions extract kiye.
Ye unit RGPV me mostly:

- Optimization techniques
- DAG
- Dead code
- Loop optimization
- Flow graph
  par focused rehta hai.

🌟 UNIT–V Topics

📌 Code Optimization

- Sources of Optimization
- Optimization of Basic Blocks
- Loops in Flow Graphs
- Dead Code Elimination
- Loop Optimization

📌 Data Flow Analysis

- Global Data Flow Analysis
- Code Improving Transformations
- Flow Graph Analysis

📌 Symbolic Debugging

🔥 MOST REPEATED TOPICS

Topic	Frequency	Importance
Dead Code Elimination	⭐⭐⭐⭐⭐	VERY HIGH
DAG Representation	⭐⭐⭐⭐⭐	VERY HIGH
Loop Optimization	⭐⭐⭐⭐	HIGH
Common Subexpression Elimination	⭐⭐⭐⭐	HIGH
Peephole Optimization	⭐⭐⭐⭐	HIGH
Basic Block Optimization	⭐⭐⭐⭐	HIGH
Flow Graph	⭐⭐⭐	HIGH
Data Flow Analysis	⭐⭐⭐	HIGH

📅 YEAR-WISE PYQ ANALYSIS

🔥 DECEMBER 2017 PAPER

✅ Q7(a)

Analyse the possible causes of dead code. Explain with an example how compiler can detect presence of dead code.

📌 Related Topic:

- Dead Code Elimination
- Optimization

⭐ MOST IMPORTANT

✅ Q7(b)

Explain:

- Common Subexpression Elimination
- Copy Propagation
- Transformation for moving loop invariant computations

📌 Related Topic:

- Loop Optimization
- Code Improving Transformations

⭐ VERY IMPORTANT

✅ Q6(b)

What is DAG? Construct DAG for the following basic block:

```
D = B * C
E = A + B
B = B * C
A = E - D
```

📌 Related Topic:

- DAG Representation
- Basic Block Optimization

⭐ MOST REPEATED

🔥 JUNE 2020 PAPER

✅ Q8(a)

Write short notes on:

- Local and Loop Optimization
- Peephole Optimization
- Dead Code Elimination

📌 Related Topic:

- Loop Optimization
- Peephole Optimization
- Dead Code Elimination

⭐ VERY IMPORTANT

✅ Q7

Construct DAG for the following basic block:

```
a = b + c
b = b - d
c = c + d
e = b + c
```

📌 Related Topic:

- DAG Representation

⭐ REPEATED AGAIN

🔥 DECEMBER 2020 PAPER

✅ Questions indirectly related to optimization:

- Flow graph analysis
- Optimization concepts
- Intermediate optimization

📌 Related Topic:

- Flow Graph
- Optimization Techniques

⭐ IMPORTANT

🔥 FINAL ANALYSIS (MOST IMPORTANT QUESTIONS)

⭐ TOP 7 MOST IMPORTANT QUESTIONS

1️⃣ Explain Dead Code Elimination

🔥 Repeated directly
🔥 Highest probability

2️⃣ Construct DAG for Basic Block

🔥 Asked multiple times

3️⃣ Explain Loop Optimization

🔥 Repeated optimization topic

4️⃣ Explain Common Subexpression Elimination

🔥 Frequently repeated

5️⃣ Explain Peephole Optimization

🔥 Important short + long question

6️⃣ Explain Code Improving Transformations

🔥 High probability

7️⃣ Explain Basic Block Optimization

🔥 Important theory question

🎯 MOST IMPORTANT SMALL TOPICS

Topic	Chance
Flow Graph	HIGH
Copy Propagation	HIGH
Constant Folding	HIGH
Loop Invariant Code Motion	HIGH
Data Flow Analysis	HIGH
Symbolic Debugging	MEDIUM

Topic	Chance
Global Optimization	HIGH

🔥 MOST EXPECTED QUESTIONS FOR UPCOMING EXAM

⭐ VERY HIGH CHANCE

1. Explain Dead Code Elimination.
2. Construct DAG for Basic Block.
3. Explain Loop Optimization.
4. Explain Peephole Optimization.
5. Explain Common Subexpression Elimination.

⭐ HIGH CHANCE

6. Explain Code Improving Transformations.
7. Explain Optimization of Basic Blocks.
8. Explain Data Flow Analysis of Flow Graph.
9. Explain Global Data Flow Analysis.

⭐ MEDIUM CHANCE

10. Explain Symbolic Debugging of Optimized Code.
11. Explain Sources of Optimization.
12. Explain Copy Propagation.

⚡ SMART STUDY STRATEGY FOR UNIT–V

🌟 MUST STUDY FIRST

✅ Dead Code Elimination
✅ DAG Representation
✅ Loop Optimization
✅ Peephole Optimization

🌟 THEN STUDY

✅ Common Subexpression Elimination
✅ Code Improving Transformations
✅ Basic Block Optimization

🌟 LAST

✅ Data Flow Analysis
✅ Symbolic Debugging
✅ Flow Graph

🔥 ONE NIGHT REVISION PRIORITY

1. Dead Code Elimination
2. DAG Representation
3. Loop Optimization

4. Peephole Optimization
5. Common Subexpression Elimination
6. Code Improving Transformations
7. Basic Block Optimization
8. Data Flow Analysis
9. Flow Graph