router.post("/new/:testId", verifyToken, async (req, res) => {

  try {

    //Check permission

    if (

      !(

        req.body.verifyAccount.role === ROLES.ADMIN || **(1)**

        req.body.verifyAccount.role === ROLES.CREATOR **(2)**

      )

    ) {

      return res **(3)**

        .status(401)

        .json({ success: false, message: "Permission denied" });

    }

    if (!req.body) **(4)**

      res.status(400).json({ **(5)**

        success: false,

        message: "Body request not found",

      });

    //Create new question

    let question = new Question({ **(6)**

      order: req.body.order,

      content: req.body.content,

      type: req.body.type,

      answers: req.body.answers,

      correctAnswers: req.body.correctAnswers,

      embededMedia: req.body.embededMedia,

    });

    //Send to Database

    question = await question.save(); **(7)**

    //TODO: Updating for Test Collection : Question list

    let test = await Test.findById(req.params.testId); **(8)**

    console.log(test);

    if (test) { **(9)**

      test.questions.push(question.id.toString()); **(10)**

      test = await Test.findOneAndUpdate({ \_id: test.id }, test, { new: true }); **(11)**

    }

    res.json({ **(12)**

      success: true,

      message: "Question created successfully",

      question: question,

    });

  } catch (error) {

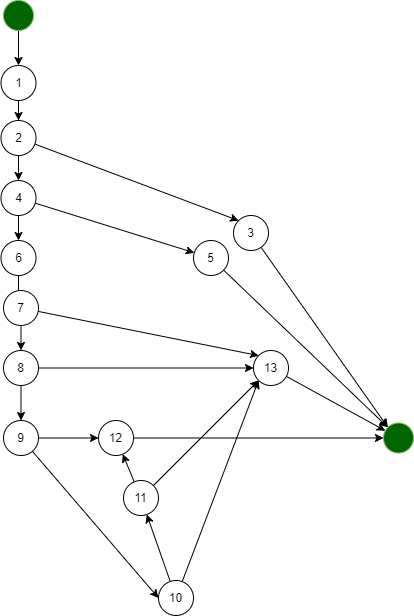
    console.log(error);

    res.status(500).json({ success: false, message: "Internal server error" }); **(13)**

  }

});

1. Create control flow graph



2. Calculate Cyclomatic complexity to identify independent linear execution paths.

Có 4 nút nhị phân => C = 7 + 1 = 8

3. Design test cases for each path

**Đường 1-> 2 -> 3 -> 13**

Pass:

(req.body.verifyAccount.role = ROLES.USER) : kết quả là

return res **(3)**

        .status(401)

        .json({ success: false, message: "Permission denied" });

**Đường 1 -> 2 -> 4 -> 5**

Pass: req.body.verifyAccount.role = ROLES.ADMIN , req.body = null

Kết quả:

      res.status(400).json({

        success: false,

        message: "Body request not found",

      });

**Đường 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 12**

Pass: req.body.verifyAccount.role = ROLES.ADMIN , req.body != null, test = null

 Kết quả

res.json({ **(12)**

      success: true,

      message: "Question created successfully",

      question: question})

**Đường 1 -> 2 -> 4 -> 6 -> 7 -> 13**

req.body.verifyAccount.role = ROLES.ADMIN , req.body != null

Bị trùng các field có giá trị là duy nhất thì xảy ra Exception

**Đường 1 -> 2 -> 4 -> 6 -> 7 --> 8 -> 13**

req.body.verifyAccount.role = ROLES.ADMIN , req.body != null

Lỗi mongodb thì xảy ra Exception

**Đường 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 10 -> 11 ->12**

Pass: req.body.verifyAccount.role = ROLES.ADMIN , req.body != null, test = null

 Kết quả

res.json({ **(12)**

      success: true,

      message: "Question created successfully",

      question: question})

**Đường 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 10 -> 13**

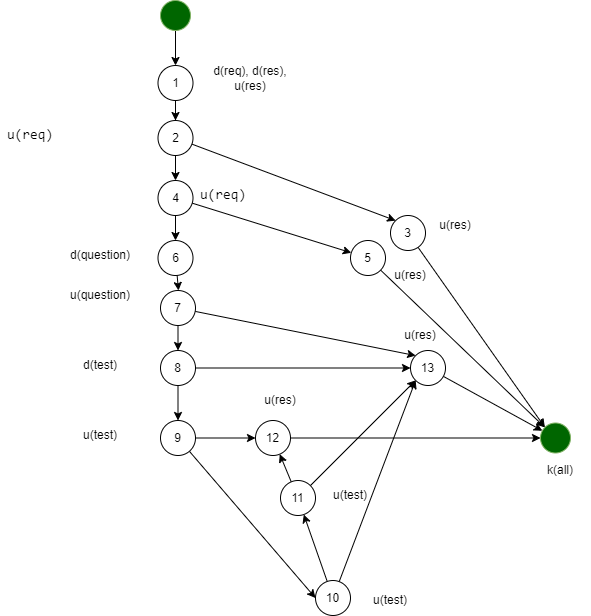
req.body.verifyAccount.role = ROLES.ADMIN , req.body != null, test != null ,question.id = null

Thì xảy ra exception

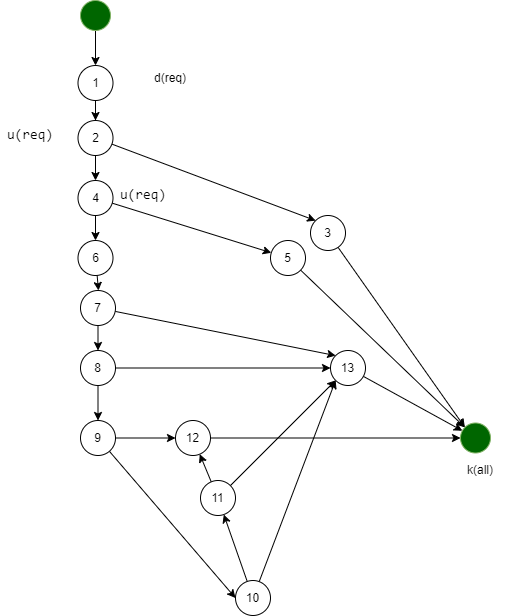
**Đường 1 -> 2 -> 4 -> 6 -> 7 -> 8 -> 9 -> 10 -> 11 ->13**

req.body.verifyAccount.role = ROLES.ADMIN , req.body != null, test != null ,question.id != null,

Lỗi mongodb thì xảy ra Exception

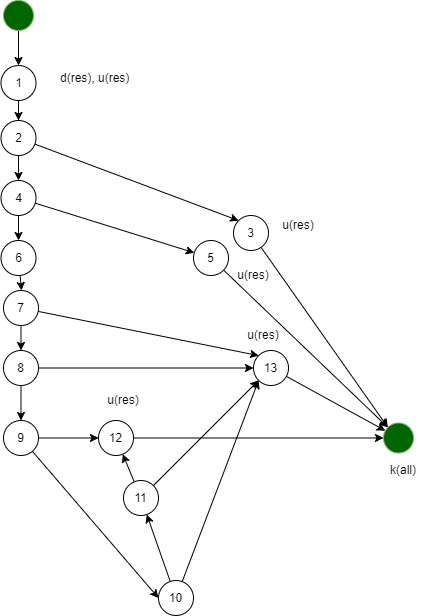


**Biến req**



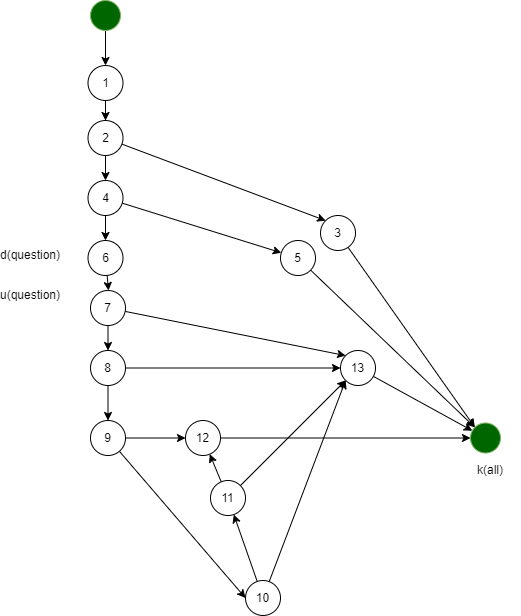
Các dòng duk và duuk đều không chứa cặp đôi nào hoạt động bất thường

**Biến res**



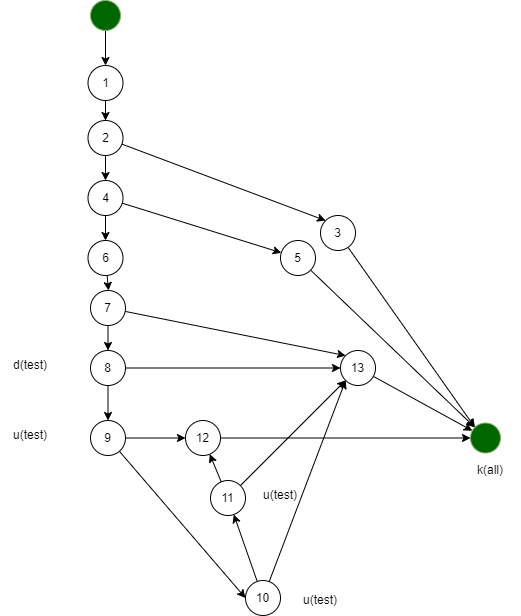
Các dòng duuk đều không chứa cặp đôi nào hoạt động bất thường.

**Biến question**

****

Các dòng duk đều không chứa cặp đôi nào bất thường.

**Biến test**

****

Các dòng dk, duk, duuk đều không chứa cặp đôi bất thường.