

# HugeGraph 异常情况记录

本文记录了对 HugeGraph 进行测试时遇到的一些没有找到解决方法的问题

## 1. 最短路径结果错误

在测试中发现，HugeGraph 官方提供的最短路径算法部分结果不正确，具体情况本文提供了两个示例说明。

算法来源：

<https://hugegraph.github.io/hugegraph-doc/clients/restful-api/traverser.html>

### 3.2.8 最短路径

实现示例：

GET [http://localhost:8080/graphs/{graph}/traversers/shortestpath?](http://localhost:8080/graphs/{graph}/traversers/shortestpath?source=1:marko&target=2:ripple&max_depth=3)  
[source="1:marko"&target="2:ripple"&max\\_depth=3](http://localhost:8080/graphs/{graph}/traversers/shortestpath?source=1:marko&target=2:ripple&max_depth=3)

### 1.1 示例一

点 A 1:1751939 到点 B 1:4165129 存在一条OUT方向的最短路径，长度为2。

The screenshot shows a REST client interface with the following details:

- Request Method:** GET
- URL:** `http://192.168.20.28:8080/graphs/g500d/traversers/shortestpath?source=%221%3A1751939%22&target=%221%3A4165129%22&max_depth=7&direction=OUT`
- Params:**

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> source	%221%3A1751939%22	
<input checked="" type="checkbox"/> target	%221%3A4165129%22	
<input checked="" type="checkbox"/> max_depth	7	
<input checked="" type="checkbox"/> direction	OUT	
Key	Value	Description
- Body:**

```
{  "path": [    "1:1751939",    "1:2698689",    "1:4165129"  ]}
```
- Status:** 200 OK, Time: 14 ms, Size: 169 B

点 B 1:4165129 到点 C 1:507269 存在一条OUT方向的最短路径，长度为1。

lear all

GET http://192.168.20.28:8080/graphs/g500d/traversers/shortestpath?source=%221%3A4165129%22&target=%221%3A507269%22

Params Authorization Headers (7) Body Pre-request Script Tests Settings

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> source	%221%3A4165129%22	
<input checked="" type="checkbox"/> target	%221%3A507269%22	
<input checked="" type="checkbox"/> max_depth	7	
<input checked="" type="checkbox"/> direction	OUT	
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 166 ms Size: 156 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "path": [
3     "1:4165129",
4     "1:507269"
5   ]
6 }
```

但是查询起始点 A 1:1751939 到终止点 C 1:507269 的最短路径时，却显示没有结果。在其他图数据库如 Galaxybase 和 Neo4j 中，该路径查询结果正常。

Untitled Request BUILD

GET http://192.168.20.28:8080/graphs/g500d/traversers/shortestpath?source=%221%3A1751939%22&target=%221%3A507269%22

Send Save

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> source	%221%3A1751939%22	
<input checked="" type="checkbox"/> target	%221%3A507269%22	
<input checked="" type="checkbox"/> max_depth	7	
<input checked="" type="checkbox"/> direction	OUT	
Key	Value	Description

Body Cookies Headers (3) Test Results Status: 200 OK Time: 1 m 11.82 s Size: 134 B Save Responses

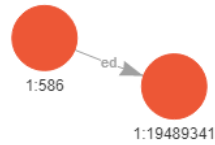
Pretty Raw Preview Visualize JSON

```
1 {
2   "path": []
3 }
```

## 1.2 示例二

点 A 1:586 到点 B 1:19489341 两点之间存在一条边直接相连，理论上最短路径长度应该为 1，但查询结果最短路径长度为3，经过了一个中间节点 1:13。

```
g.V('1:586').outE().as('r').inV().hasId('1:19489341').select('r')
```



## 2. 边数量超过1万的点无法删除

## Gremlin



[16402]

## Gremlin

```
g.V('9:10995116339640').drop()
```

### Error!

Edges size has reached tx capacity 10000

## 3. K-hop neighbor 查询抛出异常: Too many records

在 Twitter-2010 图上进行 k-hop neighbor 测试时抛出了 Too many records(must <= 800000) 异常, 从异常信息中推断, 是因为在进行扩展时某个步骤的记录数量超过了 800000, 查阅 HugeGraph 公开文档, 并未找到对于该问题的相关描述, 也未找到相关的配置项。

```
java.util.concurrent.ExecutionException: com.baidu.hugegraph.exception.LimitExceedException: Too many records(must <= 800000) for the query: Query for VERTEX where id in [1:12, 1:13, 1:15, 1:20, 1:21, 1:22, 1:31, 1:52, 1:53, 1:57, 1:58, 1:59, 1:60, 1:70, 1:87, 1:89, 1:92, 1:107, 1:108, 1:150, 1:209, 1:238, 1:242, 1:257, 1:259, 1:266, 1:277, 1:279, 1:288, 1:294, 1:295, 1:314, 1:338, 1:341, 1:350...
    at java.util.concurrent.FutureTask.report(FutureTask.java:122)
    at java.util.concurrent.FutureTask.get(FutureTask.java:192)
    at com.galaxybase.benchmark.common.test.AbstractTest.testFinish(AbstractTest.java:126)
    at com.galaxybase.benchmark.common.test.AbstractTest.startTest(AbstractTest.java:94)
    at com.galaxybase.benchmark.common.util.TestManager.doStartTest(TestManager.java:60)
    at com.galaxybase.benchmark.common.util.TestManager.startTest(TestManager.java:25)
    at com.galaxybase.benchmark.hugegraph.AutoMain.main(AutoMain.java:11)
Caused by: com.baidu.hugegraph.exception.LimitExceedException: Too many records(must <= 800000) for the query: Query for VERTEX where id in [1:12, 1:13, 1:15, 1:20, 1:21, 1:22, 1:31, 1:52, 1:53, 1:57, 1:58, 1:59, 1:60, 1:70, 1:87, 1:89, 1:92, 1:107, 1:108, 1:150, 1:209, 1:238, 1:242, 1:257, 1:259, 1:266, 1:277, 1:279, 1:288, 1:294, 1:295, 1:314, 1:338, 1:341, 1:350...
    at com.baidu.hugegraph.exception.ServerException.fromResponse(ServerException.java:44)
    at com.baidu.hugegraph.client.RestClient.checkStatus(RestClient.java:106)
    at com.baidu.hugegraph.rest.RestClient.post(RestClient.java:156)
    at com.baidu.hugegraph.rest.RestClient.post(RestClient.java:133)
    at com.baidu.hugegraph.api.gremlin.GremlinAPI.post(GremlinAPI.java:41)
    at com.baidu.hugegraph.driver.GremlinManager.execute(GremlinManager.java:51)
    at com.baidu.hugegraph.api.gremlin.GremlinRequestBuilder.execute(GremlinRequest.java:55)
    at com.galaxybase.benchmark.hugegraph.database.HugeHugeConnect.executeGremlin(Huge.java:99)
    at com.galaxybase.benchmark.hugegraph.test.util.HugeTest.onStartTest(HugeTest.java:28)
    at com.galaxybase.benchmark.hugegraph.test.util.HugeTest.onStartTest(HugeTest.java:12)
    at com.galaxybase.benchmark.common.test.AbstractTest$1.run(AbstractTest.java:85)
    at com.galaxybase.benchmark.common.test.AbstractTest$TaskCallable.call(AbstractTest.java:236)
    at java.util.concurrent.FutureTask.run(FutureTask.java:266)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
    at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
    at java.lang.Thread.run(Thread.java:748)
```