Calculate Maximum Flow

In [28]:
```
class Vertex:
    def __init__(self, name, source=False, sink=False):
        self.name = name
        self.source = source
        self.sink = sink
```

In [29]:
```
class Edge:
    def __init__(self, start, end, capacity):
        self.start = start
        self.end = end
        self.capacity = capacity
        self.flow = 0
        self.returnEdge = None
```

In [30]:
```
class FlowNetwork:
    def __init__(self):
        self.vertices = []
        self.network = {}

    def getSource(self):
        for vertex in self.vertices:
            if vertex.source == True:
                return vertex
        return None

    def getSink(self):
        for vertex in self.vertices:
            if vertex.sink == True:
                return vertex
        return None

    def getVertex(self, name):
        for vertex in self.vertices:
            if name == vertex.name:
                return vertex

    def vertexInNetwork(self, name):
        for vertex in self.vertices:
            if vertex.name == name:
                return True
        return False

    def getEdges(self):
        allEdges = []
        for vertex in self.network:
            for edge in self.network[vertex]:
                allEdges.append(edge)
        return allEdges

    def addVertex(self, name, source=False, sink=False):
        if source == True and sink == True:
            return "Vertex cannot be source and sink"
        if self.vertexInNetwork(name):
            return "Duplicate vertex"
        if source == True:
            if self.getSource() != None:
                return "Source already Exists"
        if sink == True:
            if self.getSink() != None:
                return "Sink already Exists"
        newVertex = Vertex(name, source, sink)
        self.vertices.append(newVertex)
        self.network[newVertex.name] = []

    def addEdge(self, start, end, capacity):
        if start == end:
            return "Cannot have same start and end"
        if self.vertexInNetwork(start) == False:
            return "Start vertex has not been added yet"
        if self.vertexInNetwork(end) == False:
```

```
            return "End vertex has not been added yet"
        newEdge = Edge(start, end, capacity)
        returnEdge = Edge(end, start, 0)
        newEdge.returnEdge = returnEdge
        returnEdge.returnEdge = newEdge
        vertex = self.getVertex(start)
        self.network[vertex.name].append(newEdge)
        returnVertex = self.getVertex(end)
        self.network[returnVertex.name].append(returnEdge)

    def getPath(self, start, end, path):
        if start == end:
            return path
        for edge in self.network[start]:
            residualCapacity = edge.capacity - edge.flow
            if residualCapacity > 0 and not (edge, residualCapacity) in
path:
                result = self.getPath(edge.end, end, path + [(edge, resi
dualCapacity)])
                if result != None:
                    return result

    def calculateMaxFlow(self):
        source = self.getSource()
        sink = self.getSink()
        if source == None or sink == None:
            return "Network does not have source and sink"
        path = self.getPath(source.name, sink.name, [])
        i = 0
        while path != None:
            total_path = []
            for e in path:
                total_path += [e[0].start]
            flow = min(edge[1] for edge in path)
            print(str(i) + ": " + str(total_path), flow)
            i += 1
            for edge, res in path:
                edge.flow += flow
                edge.returnEdge.flow -= flow
            path = self.getPath(source.name, sink.name, [])
        return sum(edge.flow for edge in self.network[source.name])
```

Graph Defined in Question 9

In [31]:
```
c = 0
fn = FlowNetwork()
fn.addVertex('s', True, False)
fn.addVertex('t', False, True)
fn.addVertex('a')
fn.addVertex('b')
fn.addVertex('c')
fn.addVertex('d')
fn.addEdge('s', 'a', 100 + c)
fn.addEdge('a', 't', 100 + c)
fn.addEdge('s', 'b', 100 + c)
fn.addEdge('b', 't', 100 + c)
fn.addEdge('c', 'b', 1 + c) # added edge CB
fn.addEdge('s', 'c', 200 + c)
fn.addEdge('c', 't', 200 + c)
fn.addEdge('s', 'd', 200 + c)
fn.addEdge('d', 't', 200 + c)
```

In [32]:
```
vertices = [vertex.name for vertex in fn.vertices]
edges = ['%s -> %s' % (e.start, e.end) for e in fn.getEdges()]
```

In [33]:
```
fn.calculateMaxFlow()
```

```
0: ['s', 'a'] 100
1: ['s', 'b'] 100
2: ['s', 'c', 'b', 's', 'd'] 1
3: ['s', 'b', 's', 'c', 's', 'd'] 1
4: ['s', 'b', 's', 'c', 's', 'd'] 1
5: ['s', 'b', 's', 'c', 's', 'd'] 1
6: ['s', 'b', 's', 'c', 's', 'd'] 1
7: ['s', 'b', 's', 'c', 's', 'd'] 1
8: ['s', 'b', 's', 'c', 's', 'd'] 1
9: ['s', 'b', 's', 'c', 's', 'd'] 1
10: ['s', 'b', 's', 'c', 's', 'd'] 1
11: ['s', 'b', 's', 'c', 's', 'd'] 1
12: ['s', 'b', 's', 'c', 's', 'd'] 1
13: ['s', 'b', 's', 'c', 's', 'd'] 1
14: ['s', 'b', 's', 'c', 's', 'd'] 1
15: ['s', 'b', 's', 'c', 's', 'd'] 1
16: ['s', 'b', 's', 'c', 's', 'd'] 1
17: ['s', 'b', 's', 'c', 's', 'd'] 1
18: ['s', 'b', 's', 'c', 's', 'd'] 1
19: ['s', 'b', 's', 'c', 's', 'd'] 1
20: ['s', 'b', 's', 'c', 's', 'd'] 1
21: ['s', 'b', 's', 'c', 's', 'd'] 1
22: ['s', 'b', 's', 'c', 's', 'd'] 1
23: ['s', 'b', 's', 'c', 's', 'd'] 1
24: ['s', 'b', 's', 'c', 's', 'd'] 1
25: ['s', 'b', 's', 'c', 's', 'd'] 1
26: ['s', 'b', 's', 'c', 's', 'd'] 1
27: ['s', 'b', 's', 'c', 's', 'd'] 1
28: ['s', 'b', 's', 'c', 's', 'd'] 1
29: ['s', 'b', 's', 'c', 's', 'd'] 1
30: ['s', 'b', 's', 'c', 's', 'd'] 1
31: ['s', 'b', 's', 'c', 's', 'd'] 1
32: ['s', 'b', 's', 'c', 's', 'd'] 1
33: ['s', 'b', 's', 'c', 's', 'd'] 1
34: ['s', 'b', 's', 'c', 's', 'd'] 1
35: ['s', 'b', 's', 'c', 's', 'd'] 1
36: ['s', 'b', 's', 'c', 's', 'd'] 1
37: ['s', 'b', 's', 'c', 's', 'd'] 1
38: ['s', 'b', 's', 'c', 's', 'd'] 1
39: ['s', 'b', 's', 'c', 's', 'd'] 1
40: ['s', 'b', 's', 'c', 's', 'd'] 1
41: ['s', 'b', 's', 'c', 's', 'd'] 1
42: ['s', 'b', 's', 'c', 's', 'd'] 1
43: ['s', 'b', 's', 'c', 's', 'd'] 1
44: ['s', 'b', 's', 'c', 's', 'd'] 1
45: ['s', 'b', 's', 'c', 's', 'd'] 1
46: ['s', 'b', 's', 'c', 's', 'd'] 1
47: ['s', 'b', 's', 'c', 's', 'd'] 1
48: ['s', 'b', 's', 'c', 's', 'd'] 1
49: ['s', 'b', 's', 'c', 's', 'd'] 1
50: ['s', 'b', 's', 'c', 's', 'd'] 1
51: ['s', 'b', 's', 'c', 's', 'd'] 1
52: ['s', 'b', 's', 'c', 's', 'd'] 1
53: ['s', 'b', 's', 'c', 's', 'd'] 1
54: ['s', 'b', 's', 'c', 's', 'd'] 1
55: ['s', 'b', 's', 'c', 's', 'd'] 1
56: ['s', 'b', 's', 'c', 's', 'd'] 1
```

57: ['s', 'b', 's', 'c', 's', 'd'] 1
58: ['s', 'b', 's', 'c', 's', 'd'] 1
59: ['s', 'b', 's', 'c', 's', 'd'] 1
60: ['s', 'b', 's', 'c', 's', 'd'] 1
61: ['s', 'b', 's', 'c', 's', 'd'] 1
62: ['s', 'b', 's', 'c', 's', 'd'] 1
63: ['s', 'b', 's', 'c', 's', 'd'] 1
64: ['s', 'b', 's', 'c', 's', 'd'] 1
65: ['s', 'b', 's', 'c', 's', 'd'] 1
66: ['s', 'b', 's', 'c', 's', 'd'] 1
67: ['s', 'b', 's', 'c', 's', 'd'] 1
68: ['s', 'b', 's', 'c', 's', 'd'] 1
69: ['s', 'b', 's', 'c', 's', 'd'] 1
70: ['s', 'b', 's', 'c', 's', 'd'] 1
71: ['s', 'b', 's', 'c', 's', 'd'] 1
72: ['s', 'b', 's', 'c', 's', 'd'] 1
73: ['s', 'b', 's', 'c', 's', 'd'] 1
74: ['s', 'b', 's', 'c', 's', 'd'] 1
75: ['s', 'b', 's', 'c', 's', 'd'] 1
76: ['s', 'b', 's', 'c', 's', 'd'] 1
77: ['s', 'b', 's', 'c', 's', 'd'] 1
78: ['s', 'b', 's', 'c', 's', 'd'] 1
79: ['s', 'b', 's', 'c', 's', 'd'] 1
80: ['s', 'b', 's', 'c', 's', 'd'] 1
81: ['s', 'b', 's', 'c', 's', 'd'] 1
82: ['s', 'b', 's', 'c', 's', 'd'] 1
83: ['s', 'b', 's', 'c', 's', 'd'] 1
84: ['s', 'b', 's', 'c', 's', 'd'] 1
85: ['s', 'b', 's', 'c', 's', 'd'] 1
86: ['s', 'b', 's', 'c', 's', 'd'] 1
87: ['s', 'b', 's', 'c', 's', 'd'] 1
88: ['s', 'b', 's', 'c', 's', 'd'] 1
89: ['s', 'b', 's', 'c', 's', 'd'] 1
90: ['s', 'b', 's', 'c', 's', 'd'] 1
91: ['s', 'b', 's', 'c', 's', 'd'] 1
92: ['s', 'b', 's', 'c', 's', 'd'] 1
93: ['s', 'b', 's', 'c', 's', 'd'] 1
94: ['s', 'b', 's', 'c', 's', 'd'] 1
95: ['s', 'b', 's', 'c', 's', 'd'] 1
96: ['s', 'b', 's', 'c', 's', 'd'] 1
97: ['s', 'b', 's', 'c', 's', 'd'] 1
98: ['s', 'b', 's', 'c', 's', 'd'] 1
99: ['s', 'b', 's', 'c', 's', 'd'] 1
100: ['s', 'b', 's', 'c', 's', 'd'] 1
101: ['s', 'b', 's', 'c', 's', 'd'] 1
102: ['s', 'b', 's', 'c', 's', 'd'] 1
103: ['s', 'b', 's', 'c', 's', 'd'] 1
104: ['s', 'b', 's', 'c', 's', 'd'] 1
105: ['s', 'b', 's', 'c', 's', 'd'] 1
106: ['s', 'b', 's', 'c', 's', 'd'] 1
107: ['s', 'b', 's', 'c', 's', 'd'] 1
108: ['s', 'b', 's', 'c', 's', 'd'] 1
109: ['s', 'b', 's', 'c', 's', 'd'] 1
110: ['s', 'b', 's', 'c', 's', 'd'] 1
111: ['s', 'b', 's', 'c', 's', 'd'] 1
112: ['s', 'b', 's', 'c', 's', 'd'] 1
113: ['s', 'b', 's', 'c', 's', 'd'] 1

```
114: ['s', 'b', 's', 'c', 's', 'd'] 1
115: ['s', 'b', 's', 'c', 's', 'd'] 1
116: ['s', 'b', 's', 'c', 's', 'd'] 1
117: ['s', 'b', 's', 'c', 's', 'd'] 1
118: ['s', 'b', 's', 'c', 's', 'd'] 1
119: ['s', 'b', 's', 'c', 's', 'd'] 1
120: ['s', 'b', 's', 'c', 's', 'd'] 1
121: ['s', 'b', 's', 'c', 's', 'd'] 1
122: ['s', 'b', 's', 'c', 's', 'd'] 1
123: ['s', 'b', 's', 'c', 's', 'd'] 1
124: ['s', 'b', 's', 'c', 's', 'd'] 1
125: ['s', 'b', 's', 'c', 's', 'd'] 1
126: ['s', 'b', 's', 'c', 's', 'd'] 1
127: ['s', 'b', 's', 'c', 's', 'd'] 1
128: ['s', 'b', 's', 'c', 's', 'd'] 1
129: ['s', 'b', 's', 'c', 's', 'd'] 1
130: ['s', 'b', 's', 'c', 's', 'd'] 1
131: ['s', 'b', 's', 'c', 's', 'd'] 1
132: ['s', 'b', 's', 'c', 's', 'd'] 1
133: ['s', 'b', 's', 'c', 's', 'd'] 1
134: ['s', 'b', 's', 'c', 's', 'd'] 1
135: ['s', 'b', 's', 'c', 's', 'd'] 1
136: ['s', 'b', 's', 'c', 's', 'd'] 1
137: ['s', 'b', 's', 'c', 's', 'd'] 1
138: ['s', 'b', 's', 'c', 's', 'd'] 1
139: ['s', 'b', 's', 'c', 's', 'd'] 1
140: ['s', 'b', 's', 'c', 's', 'd'] 1
141: ['s', 'b', 's', 'c', 's', 'd'] 1
142: ['s', 'b', 's', 'c', 's', 'd'] 1
143: ['s', 'b', 's', 'c', 's', 'd'] 1
144: ['s', 'b', 's', 'c', 's', 'd'] 1
145: ['s', 'b', 's', 'c', 's', 'd'] 1
146: ['s', 'b', 's', 'c', 's', 'd'] 1
147: ['s', 'b', 's', 'c', 's', 'd'] 1
148: ['s', 'b', 's', 'c', 's', 'd'] 1
149: ['s', 'b', 's', 'c', 's', 'd'] 1
150: ['s', 'b', 's', 'c', 's', 'd'] 1
151: ['s', 'b', 's', 'c', 's', 'd'] 1
152: ['s', 'b', 's', 'c', 's', 'd'] 1
153: ['s', 'b', 's', 'c', 's', 'd'] 1
154: ['s', 'b', 's', 'c', 's', 'd'] 1
155: ['s', 'b', 's', 'c', 's', 'd'] 1
156: ['s', 'b', 's', 'c', 's', 'd'] 1
157: ['s', 'b', 's', 'c', 's', 'd'] 1
158: ['s', 'b', 's', 'c', 's', 'd'] 1
159: ['s', 'b', 's', 'c', 's', 'd'] 1
160: ['s', 'b', 's', 'c', 's', 'd'] 1
161: ['s', 'b', 's', 'c', 's', 'd'] 1
162: ['s', 'b', 's', 'c', 's', 'd'] 1
163: ['s', 'b', 's', 'c', 's', 'd'] 1
164: ['s', 'b', 's', 'c', 's', 'd'] 1
165: ['s', 'b', 's', 'c', 's', 'd'] 1
166: ['s', 'b', 's', 'c', 's', 'd'] 1
167: ['s', 'b', 's', 'c', 's', 'd'] 1
168: ['s', 'b', 's', 'c', 's', 'd'] 1
169: ['s', 'b', 's', 'c', 's', 'd'] 1
170: ['s', 'b', 's', 'c', 's', 'd'] 1
```

171: ['s', 'b', 's', 'c', 's', 'd'] 1
172: ['s', 'b', 's', 'c', 's', 'd'] 1
173: ['s', 'b', 's', 'c', 's', 'd'] 1
174: ['s', 'b', 's', 'c', 's', 'd'] 1
175: ['s', 'b', 's', 'c', 's', 'd'] 1
176: ['s', 'b', 's', 'c', 's', 'd'] 1
177: ['s', 'b', 's', 'c', 's', 'd'] 1
178: ['s', 'b', 's', 'c', 's', 'd'] 1
179: ['s', 'b', 's', 'c', 's', 'd'] 1
180: ['s', 'b', 's', 'c', 's', 'd'] 1
181: ['s', 'b', 's', 'c', 's', 'd'] 1
182: ['s', 'b', 's', 'c', 's', 'd'] 1
183: ['s', 'b', 's', 'c', 's', 'd'] 1
184: ['s', 'b', 's', 'c', 's', 'd'] 1
185: ['s', 'b', 's', 'c', 's', 'd'] 1
186: ['s', 'b', 's', 'c', 's', 'd'] 1
187: ['s', 'b', 's', 'c', 's', 'd'] 1
188: ['s', 'b', 's', 'c', 's', 'd'] 1
189: ['s', 'b', 's', 'c', 's', 'd'] 1
190: ['s', 'b', 's', 'c', 's', 'd'] 1
191: ['s', 'b', 's', 'c', 's', 'd'] 1
192: ['s', 'b', 's', 'c', 's', 'd'] 1
193: ['s', 'b', 's', 'c', 's', 'd'] 1
194: ['s', 'b', 's', 'c', 's', 'd'] 1
195: ['s', 'b', 's', 'c', 's', 'd'] 1
196: ['s', 'b', 's', 'c', 's', 'd'] 1
197: ['s', 'b', 's', 'c', 's', 'd'] 1
198: ['s', 'b', 's', 'c', 's', 'd'] 1
199: ['s', 'b', 's', 'c', 's', 'd'] 1
200: ['s', 'b', 's', 'c', 's', 'd'] 1
201: ['s', 'b', 's', 'c', 's', 'd'] 1
202: ['s', 'b', 's', 'c'] 1
203: ['s', 'b', 's', 'c'] 1
204: ['s', 'b', 's', 'c'] 1
205: ['s', 'b', 's', 'c'] 1
206: ['s', 'b', 's', 'c'] 1
207: ['s', 'b', 's', 'c'] 1
208: ['s', 'b', 's', 'c'] 1
209: ['s', 'b', 's', 'c'] 1
210: ['s', 'b', 's', 'c'] 1
211: ['s', 'b', 's', 'c'] 1
212: ['s', 'b', 's', 'c'] 1
213: ['s', 'b', 's', 'c'] 1
214: ['s', 'b', 's', 'c'] 1
215: ['s', 'b', 's', 'c'] 1
216: ['s', 'b', 's', 'c'] 1
217: ['s', 'b', 's', 'c'] 1
218: ['s', 'b', 's', 'c'] 1
219: ['s', 'b', 's', 'c'] 1
220: ['s', 'b', 's', 'c'] 1
221: ['s', 'b', 's', 'c'] 1
222: ['s', 'b', 's', 'c'] 1
223: ['s', 'b', 's', 'c'] 1
224: ['s', 'b', 's', 'c'] 1
225: ['s', 'b', 's', 'c'] 1
226: ['s', 'b', 's', 'c'] 1
227: ['s', 'b', 's', 'c'] 1

228: ['s', 'b', 's', 'c'] 1
229: ['s', 'b', 's', 'c'] 1
230: ['s', 'b', 's', 'c'] 1
231: ['s', 'b', 's', 'c'] 1
232: ['s', 'b', 's', 'c'] 1
233: ['s', 'b', 's', 'c'] 1
234: ['s', 'b', 's', 'c'] 1
235: ['s', 'b', 's', 'c'] 1
236: ['s', 'b', 's', 'c'] 1
237: ['s', 'b', 's', 'c'] 1
238: ['s', 'b', 's', 'c'] 1
239: ['s', 'b', 's', 'c'] 1
240: ['s', 'b', 's', 'c'] 1
241: ['s', 'b', 's', 'c'] 1
242: ['s', 'b', 's', 'c'] 1
243: ['s', 'b', 's', 'c'] 1
244: ['s', 'b', 's', 'c'] 1
245: ['s', 'b', 's', 'c'] 1
246: ['s', 'b', 's', 'c'] 1
247: ['s', 'b', 's', 'c'] 1
248: ['s', 'b', 's', 'c'] 1
249: ['s', 'b', 's', 'c'] 1
250: ['s', 'b', 's', 'c'] 1
251: ['s', 'b', 's', 'c'] 1
252: ['s', 'b', 's', 'c'] 1
253: ['s', 'b', 's', 'c'] 1
254: ['s', 'b', 's', 'c'] 1
255: ['s', 'b', 's', 'c'] 1
256: ['s', 'b', 's', 'c'] 1
257: ['s', 'b', 's', 'c'] 1
258: ['s', 'b', 's', 'c'] 1
259: ['s', 'b', 's', 'c'] 1
260: ['s', 'b', 's', 'c'] 1
261: ['s', 'b', 's', 'c'] 1
262: ['s', 'b', 's', 'c'] 1
263: ['s', 'b', 's', 'c'] 1
264: ['s', 'b', 's', 'c'] 1
265: ['s', 'b', 's', 'c'] 1
266: ['s', 'b', 's', 'c'] 1
267: ['s', 'b', 's', 'c'] 1
268: ['s', 'b', 's', 'c'] 1
269: ['s', 'b', 's', 'c'] 1
270: ['s', 'b', 's', 'c'] 1
271: ['s', 'b', 's', 'c'] 1
272: ['s', 'b', 's', 'c'] 1
273: ['s', 'b', 's', 'c'] 1
274: ['s', 'b', 's', 'c'] 1
275: ['s', 'b', 's', 'c'] 1
276: ['s', 'b', 's', 'c'] 1
277: ['s', 'b', 's', 'c'] 1
278: ['s', 'b', 's', 'c'] 1
279: ['s', 'b', 's', 'c'] 1
280: ['s', 'b', 's', 'c'] 1
281: ['s', 'b', 's', 'c'] 1
282: ['s', 'b', 's', 'c'] 1
283: ['s', 'b', 's', 'c'] 1
284: ['s', 'b', 's', 'c'] 1

285: ['s', 'b', 's', 'c'] 1
286: ['s', 'b', 's', 'c'] 1
287: ['s', 'b', 's', 'c'] 1
288: ['s', 'b', 's', 'c'] 1
289: ['s', 'b', 's', 'c'] 1
290: ['s', 'b', 's', 'c'] 1
291: ['s', 'b', 's', 'c'] 1
292: ['s', 'b', 's', 'c'] 1
293: ['s', 'b', 's', 'c'] 1
294: ['s', 'b', 's', 'c'] 1
295: ['s', 'b', 's', 'c'] 1
296: ['s', 'b', 's', 'c'] 1
297: ['s', 'b', 's', 'c'] 1
298: ['s', 'b', 's', 'c'] 1
299: ['s', 'b', 's', 'c'] 1
300: ['s', 'b', 's', 'c'] 1
301: ['s', 'b', 's', 'c'] 1
302: ['s', 'b', 's', 'c'] 1
303: ['s', 'b', 's', 'c'] 1
304: ['s', 'b', 's', 'c'] 1
305: ['s', 'b', 's', 'c'] 1
306: ['s', 'b', 's', 'c'] 1
307: ['s', 'b', 's', 'c'] 1
308: ['s', 'b', 's', 'c'] 1
309: ['s', 'b', 's', 'c'] 1
310: ['s', 'b', 's', 'c'] 1
311: ['s', 'b', 's', 'c'] 1
312: ['s', 'b', 's', 'c'] 1
313: ['s', 'b', 's', 'c'] 1
314: ['s', 'b', 's', 'c'] 1
315: ['s', 'b', 's', 'c'] 1
316: ['s', 'b', 's', 'c'] 1
317: ['s', 'b', 's', 'c'] 1
318: ['s', 'b', 's', 'c'] 1
319: ['s', 'b', 's', 'c'] 1
320: ['s', 'b', 's', 'c'] 1
321: ['s', 'b', 's', 'c'] 1
322: ['s', 'b', 's', 'c'] 1
323: ['s', 'b', 's', 'c'] 1
324: ['s', 'b', 's', 'c'] 1
325: ['s', 'b', 's', 'c'] 1
326: ['s', 'b', 's', 'c'] 1
327: ['s', 'b', 's', 'c'] 1
328: ['s', 'b', 's', 'c'] 1
329: ['s', 'b', 's', 'c'] 1
330: ['s', 'b', 's', 'c'] 1
331: ['s', 'b', 's', 'c'] 1
332: ['s', 'b', 's', 'c'] 1
333: ['s', 'b', 's', 'c'] 1
334: ['s', 'b', 's', 'c'] 1
335: ['s', 'b', 's', 'c'] 1
336: ['s', 'b', 's', 'c'] 1
337: ['s', 'b', 's', 'c'] 1
338: ['s', 'b', 's', 'c'] 1
339: ['s', 'b', 's', 'c'] 1
340: ['s', 'b', 's', 'c'] 1
341: ['s', 'b', 's', 'c'] 1

342: ['s', 'b', 's', 'c'] 1
343: ['s', 'b', 's', 'c'] 1
344: ['s', 'b', 's', 'c'] 1
345: ['s', 'b', 's', 'c'] 1
346: ['s', 'b', 's', 'c'] 1
347: ['s', 'b', 's', 'c'] 1
348: ['s', 'b', 's', 'c'] 1
349: ['s', 'b', 's', 'c'] 1
350: ['s', 'b', 's', 'c'] 1
351: ['s', 'b', 's', 'c'] 1
352: ['s', 'b', 's', 'c'] 1
353: ['s', 'b', 's', 'c'] 1
354: ['s', 'b', 's', 'c'] 1
355: ['s', 'b', 's', 'c'] 1
356: ['s', 'b', 's', 'c'] 1
357: ['s', 'b', 's', 'c'] 1
358: ['s', 'b', 's', 'c'] 1
359: ['s', 'b', 's', 'c'] 1
360: ['s', 'b', 's', 'c'] 1
361: ['s', 'b', 's', 'c'] 1
362: ['s', 'b', 's', 'c'] 1
363: ['s', 'b', 's', 'c'] 1
364: ['s', 'b', 's', 'c'] 1
365: ['s', 'b', 's', 'c'] 1
366: ['s', 'b', 's', 'c'] 1
367: ['s', 'b', 's', 'c'] 1
368: ['s', 'b', 's', 'c'] 1
369: ['s', 'b', 's', 'c'] 1
370: ['s', 'b', 's', 'c'] 1
371: ['s', 'b', 's', 'c'] 1
372: ['s', 'b', 's', 'c'] 1
373: ['s', 'b', 's', 'c'] 1
374: ['s', 'b', 's', 'c'] 1
375: ['s', 'b', 's', 'c'] 1
376: ['s', 'b', 's', 'c'] 1
377: ['s', 'b', 's', 'c'] 1
378: ['s', 'b', 's', 'c'] 1
379: ['s', 'b', 's', 'c'] 1
380: ['s', 'b', 's', 'c'] 1
381: ['s', 'b', 's', 'c'] 1
382: ['s', 'b', 's', 'c'] 1
383: ['s', 'b', 's', 'c'] 1
384: ['s', 'b', 's', 'c'] 1
385: ['s', 'b', 's', 'c'] 1
386: ['s', 'b', 's', 'c'] 1
387: ['s', 'b', 's', 'c'] 1
388: ['s', 'b', 's', 'c'] 1
389: ['s', 'b', 's', 'c'] 1
390: ['s', 'b', 's', 'c'] 1
391: ['s', 'b', 's', 'c'] 1
392: ['s', 'b', 's', 'c'] 1
393: ['s', 'b', 's', 'c'] 1
394: ['s', 'b', 's', 'c'] 1
395: ['s', 'b', 's', 'c'] 1
396: ['s', 'b', 's', 'c'] 1
397: ['s', 'b', 's', 'c'] 1
398: ['s', 'b', 's', 'c'] 1

```
          399: ['s', 'b', 's', 'c'] 1
          400: ['s', 'b', 's', 'c'] 1
          401: ['s', 'b', 'c'] 1
```

```
Out[33]: 600
```

As you can see, this edge still causes the algorithm to run 402 iterations.